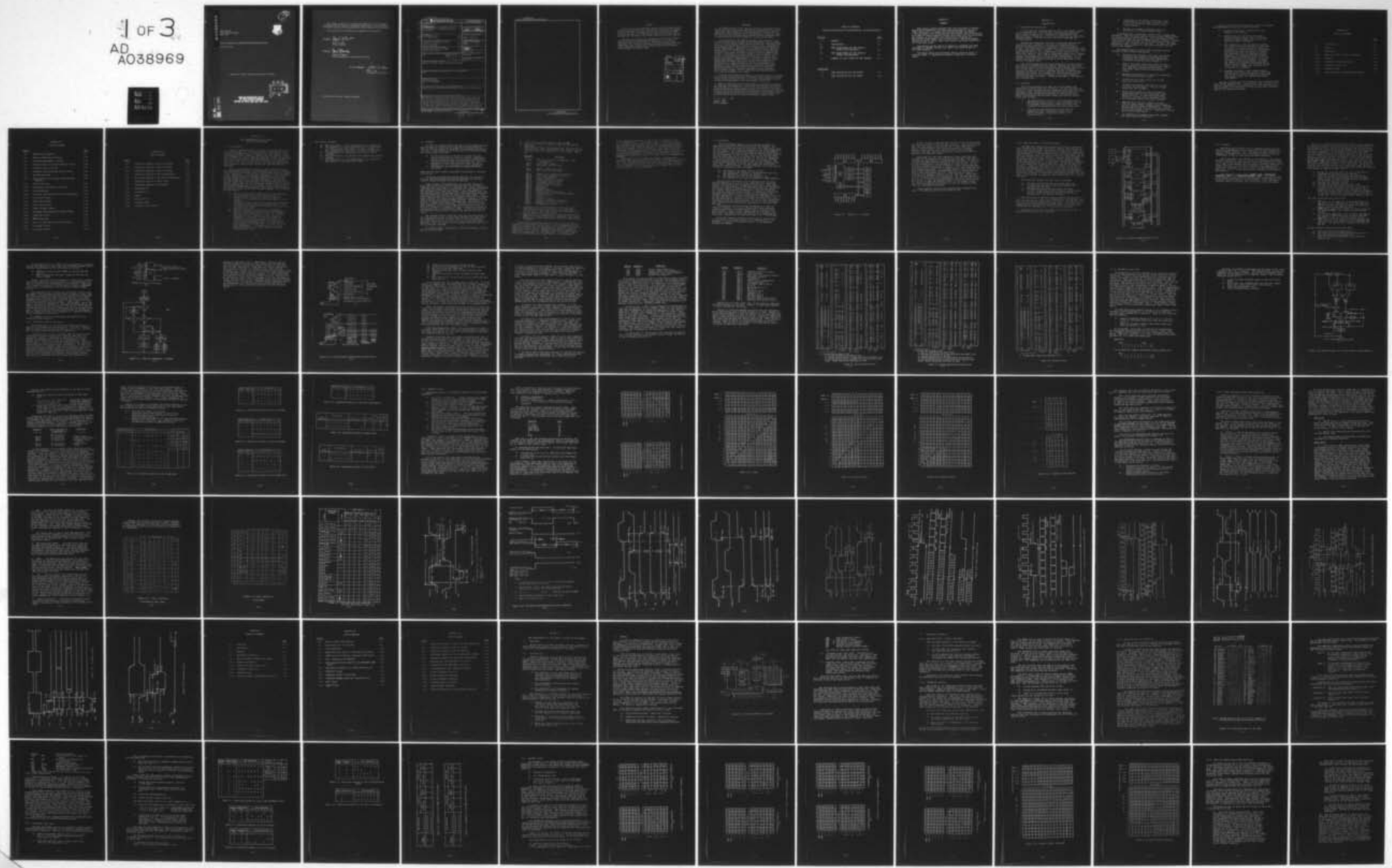
AD-A038 969 GENERAL ELECTRIC CO PITTSFIELD MASS ORDNANCF SYSTEMS F/G 9/5
DIGITAL MICROCIRCUIT CHARACTERIZATION AND SPECIFICATION.(U)
MAR 77 T M OSTROWSKI F30602-74-C-0159

UNCLASSIFIED RADC-TR-77-91 NL

1 OF 3
AD
A038969



12

NW

RADC-TR-77-91
Final Technical Report
March 1977

DIGITAL MICROCIRCUIT CHARACTERIZATION AND SPECIFICATION

General Electric

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

1473

This report contains a few illustrations which are not of the highest printing quality but because of economical consideration, it was determined in the best interest of the government that they be used in this publication.

This report has been reviewed and is approved for publication.

APPROVED: Regis C Hilow

REGIS C. HILOW
Project Engineer

APPROVED: Joseph J Naresky

JOSEPH J. NARESKY
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

18 1. REPORT NUMBER RADC-TR-77-91	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
6 4. TITLE (and Subtitle) DIGITAL MICROCIRCUIT CHARACTERIZATION AND SPECIFICATION,		9 5. TYPE OF REPORT & PERIOD COVERED Final Technical Report, Apr[redacted]975 — Dec[redacted]976
		6. PERFORMING ORG. REPORT NUMBER N/A
10 7. AUTHOR(s) Thomas M. Ostrowski		15 8. CONTRACT OR GRANT NUMBER(s) F30602-74-C-0159
9. PERFORMING ORGANIZATION NAME AND ADDRESS General Electric/Ordanance Systems 100 Plastics Avenue Pittsfield MA 01201		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 16 55190433 17 94
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBRM) Griffiss AFB NY 13441		11 12. REPORT DATE Mar[redacted]977
		13. NUMBER OF PAGES 259
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same 12 259p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Regis C. Hilow (RBRM)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Microprocessors, test philosophy, MIL-M38510 slash sheets, functional tests

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objective of this effort was to generate MIL-M-38510 slash sheets for microprocessor integrated circuits. The report includes the general microprocessor test philosophy developed and used, technical analyses of manufacturer generated functional test programs, changes made to manufacturer programs, and the approach taken in slash sheet development. Two slash sheets were developed; /400 for the 6800 and /420 for the 8080A microprocessor (not included in the report). Also included are analyses of functional tests for many low power Schottky flip flops and one CMOS register circuit (CD4035A).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

149 175

mt

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This Final Report was prepared by General Electric Ordnance Systems, 100 Plastics Avenue, Pittsfield, Massachusetts, for Rome Air Development Center, Griffiss Air Force Base, New York 13441 under Contract F30602-74-C-0159, Job Order No. 55190433 which covers the period April 1975 to December 1976. Mr. Regis C. Hilow, RADC (RBRM) was the Project Engineer.

The work on this project was performed by the Electronic Circuits Engineering Operation and Components Engineering Unit. Project responsibility was held by Mr. Thomas M. Ostrowski of Circuit Design Engineering. Key individuals who made significant contributions to this report were Messrs. Lawrence DeLuca, Daniel Weidman, David Prystasz, Charles Shepard, and Clarence Carey.

EVALUATION

The prime objective of this study was to review or prepare MIL-M-38510 digital integrated circuit detail specifications assuring technical accuracy, completeness, and conformance to established military standards. Special emphasis was placed on state-of-the-art microprocessors to establish a baseline electrical test philosophy and to determine the most economical means for guaranteeing the electrical integrity of a given microprocessor.

Two microprocessors, the 8080A and the 6800, were selected for study based on their present or projected use in DOD equipment. With the cooperation of the device vendors, General Electric pursued this investigation well beyond the allocated contract resources to assure accurate, comprehensive and meaningful results. The end item of this endeavor are two JAN specifications, M38510/400 and M38510/420, covering the two previously mentioned microprocessors. A test philosophy for microprocessors is developed and presented in this report and forms the basis for these two M38510 specifications. A new meaning to "testing confidence level" as applied to microprocessors is also presented, hopefully clearing up the confusion that exists in the amount of testing required. The study of these two devices surfaced many important problem areas, not previously considered in the JAN specification system, that must be treated in the development of specifications for complex LSI devices, such as microprocessors. Most important of all, it became evident early in this effort that maximum cooperation between the device vendor and government or contractor specification writer is a mandatory requirement to produce meaningful LSI procurement documents. Details describing these problem areas and their scope are presented in this report.

In general, this study was considered to be very successful in accomplishing the initially established goals. The General Electric personnel assigned to this effort, as in past contracts of this type, have performed in an outstanding professional manner. This report attests to this fact.

RADC, as preparing activity of MIL-M-38510, is responsible for managing the development and preparation of detail slash sheets for this specification. This study and future studies of this type will be continued to assure that the MIL-M-38510 detail specifications are electrically accurate, cost effective, timely and capable of guaranteeing that microcircuits will perform as specified in design applications. Efforts will be expanded to include assurance that microcircuits will perform to a specified reliability figure as well.

REGIS C. HILOW
Project Engineer

TABLE OF CONTENTS

DIGITAL MICROCIRCUIT CHARACTERIZATION AND SPECIFICATION

SECTION I

SUMMARY

This report details analyses that were performed on several manufacturer-generated functional specifications for two 8-bit fixed instruction, n-channel microprocessors,(6800 and 8080A). A microprocessor test philosophy was established, and modifications were made to these programs to make them comply with the philosophy. The results of this effort were then used to generate two MIL-M-38510 slash sheets (not included here). The slash sheets include DC and AC tests, worst-case timing conditions, waveforms and functional tests.

One section of the report is devoted to analyses that were performed on SSI/MSI devices (all LSTTL flip flops except for one CMOS register).

Functional tests were generated using a stuck at logic "1", stuck at logic "0" approach to achieve a 100% test confidence level.

SECTION II

INTRODUCTION

2.1 Objective and Background

At the time this characterization effort was started, except for IC manufacturers' in-house test procedures and large automatic tester manufacturers' programs, no comprehensive test specifications, procedures or test philosophies for microprocessor testing existed. This was true for both the commercial and military market.

Perhaps the commitment involved in examining existing procedures and philosophies for accuracy and completeness or generating them anew made microprocessor testing an unknown entity. The objective was to establish a test philosophy and then to apply that philosophy to develop a military microprocessor slash sheet (MIL-M-38510). This slash sheet should include a full functional test, all the normally referenced static tests (leakage, output logic levels, etc.) and AC timing measurements. For all testing, worst-case timing, clocking and loading should be considered and theresultant specification should guarantee a high Testing Confidence Level.

The characterization effort was centered on two 8-bit fixed instruction, n-channel devices, namely, the 6800 and the 8080A. This report does not include the actual slash sheets (MIL-M-38510/-400 for the 6800, /420 for the 8080A) that were developed. Included are several reports regarding analyses of manufacturer-generated functional test programs for microprocessors and the modifications made to these programs to make them compatible with the new test philosophy. Timing and parametric measurements are discussed also.

2.2 Approach

Since microprocessors are made up of many digital MSI semiconductor memory and logic functions, the test problems are similar to those encountered with subsystem test (many functions, high gate count, limited visibility/accessibility) and those experienced with the MSI/LSI semiconductor memory functions. Before functionally characterizing these LSI devices, several areas were reviewed.

1) The applicability of S-A-1, S-A-0 and semiconductor memory test philosophies to microprocessors relative to probable failure modes, number of tests, and economics of generating and performing these tests.

2) Test development based upon a model which is composed of small functional blocks such as ALU's, registers, scratch pad memory, I/O circuitry etc.

3) Comparison of the effect of different logic mechanizations of the same functions on the device specification, test generation and implementation.

4) The use of a dynamic functional test as a substitute for complex dynamic tests.

In establishing a philosophy, several test programs were evaluated. They were being used by microprocessor manufacturers for microprocessor product testing. The conclusion of each evaluation was that the test programs were not complete. Tabular "check lists" were developed to act as indicators of the completeness of each of these test programs. In almost all cases subfunction tests (register or ALU) were scattered over the entire test program.

The approach used to develop the functional test is outlined in the following philosophy.

1) Partition the processor into basic functional blocks such as registers, multiplexers and arithmetic and logic functions, and generate a detailed functional block diagram.

2) Each of the basic functional blocks is then tested using proven test patterns which result in a high Test Confidence Level (TCL). In some cases these blocks are exhaustively tested.

3) Generate test patterns to verify the integrity of the data and control paths.

4) Verify that all instructions perform the specified operations.

5) Include test patterns that check for known processor sensitivities. This may include vendor and user feedback.

6) The AC and DC tests are then performed by integrating them with the functional tests where possible. In some cases additional vectors or patterns may be required to set up worst-case control and timing criteria.

7) Replication of the test patterns is then done for power supply voltage, clock frequency and for special timing considerations. The test pattern may require five replications in order to fully worst-case a device. Temperature testing adds further replication.

8) All tests which include the functional, dynamic and static tests are verified.

The approach used in specifying the static and dynamic characteristics is outlined in the following:

1) Obtain an electrical schematic and logic diagram of the device.

2) Generate DC tests including input and output leakage currents, input protection circuitry clamp voltage, and current and input and output capacitance.

3) The complexity of microprocessors requires that some instruction sequence be performed in order to determine if device inputs are responding to minimum threshold requirements. Therefore the input threshold test levels are integrated with the dynamic waveforms. Similarly, the minimum acceptable output logic levels are measured during or at the end of an instruction sequence. Currently non-bipolar devices are dynamic in nature and cannot be completely tested statically. For the bipolar types some liberties may be taken such as stopping the processor clock at the end of a sequence.

4) Generate the dynamic test criteria which include worst-case clock phasing, worst-case set up and hold-time criteria for input data and control inputs and input/output delay requirements.

The test patterns that were generated also considered automatic tester compatibility. Since testers vary in their architecture, capabilities and programmability, the implementation will be easier (more difficult) or less (more) time consuming to perform on a particular tester.

SECTION III

TABLE OF CONTENTS

SECTION III

LIST OF FIGURES

SECTION III

LIST OF TABLES

SECTION III

TEST DEVELOPMENT FOR THE VENDOR A Mc6800 MICROPROCESSOR

3.1 Introduction

Vendor A developed a set of vectors (Functional Test Set I) which was intended for production testing of the Mc6800 microprocessor and debugging any design errors, particularly in the microprograms (PLA). Vendor A stated that this test is longer than necessary for fault detection purposes. Since the development of the chip, Vendor A has been examining both in-house and field failures, to determine which failures are not caught by Test Set I. In order to shorten the test set and detect these other failures a new test set (Functional Test Set II) was developed.

Due to the complexity of this and other VLSI (Very Large Scale Integration) devices, the development and use of all possible input vectors for each state of the machine would be impractical. Therefore, for either test generation or test evaluation, a model representing the internal functional blocks of the uP (microprocessor) must be used which identifies the internal machine states and interconnecting data paths. This should include a gate-level logic diagram, timing diagram and a block diagram which shows the major functional areas of the uP and the interconnecting data and control paths. Since neither a logic diagram nor a sufficiently detailed block diagram was available, the approach taken to evaluate the effectiveness of the functional test was to:

1) Section the uP into functional blocks and develop a block diagram based on available information.
2) Contact Vendor A personnel regarding internal operation of the Mc6800 and the correctness of this diagram.
3) Analyze the test vectors to determine what rationale was used and the validity and completeness of this rationale with respect to verification of the blocks and instruction set.
4) Discuss the evaluation with Vendor A personnel.
5) Based on experience, evaluate the effectiveness of the vectors in comparison with the type of tests required for each functional block. In order to evaluate the test vectors more effectively and efficiently, a computer program was written to convert the binary representation of the vectors to an assembler language level of representation. This program will be referred to in this report as the "disassembler". It determined the op codes being executed and also the addresses, data and control signals used.

This report details:

1) The development of the microprocessor test philosophy.
2) The evaluation of Functional Test Set II to determine its effectiveness with respect to test criteria. It also compares this test set to "Functional Test Set I".
3) The development of additional test vectors where required.
4) The specification of waveforms and sequence for applied signals.
5) The development of a new MIL-M-38510 Logic Integrity Test (LIT) or functional test format.

3.2 Summary

In order to evaluate these test vectors the Mc6800 function was sectored into functional blocks with observable states. After an initial review of the Mc6800 block diagram and instruction set, it was determined that the function could be sectored into three broad categories:

1) the Machine Control area which includes function areas such as timing, interrupt control, instruction decode, branching logic and peripheral interfacing;
2) the Arithmetic sections including the ALU, one's complementor, shifter and condition detectors (zero, negative etc. for arithmetic and logic operation);
3) the Registers and associated Increment/Decrement Circuitry.

These sectors were further subdivided as necessary to evaluate the vectors.

The Vendor A interpreter program was then run on the op codes of the disassembled program to determine the internal register states and data flow through the ALU.

In the process of disassembling, two op codes (DC_{16} and FC_{16}) were found which are not listed by Vendor A as legal op codes and were not found in the previous set. Vendor A personnel had previously stated that no additional op codes existed for test purposes or for any other reason. When presented with this information neither the designers nor those presently associated with the 6800 could explain the reason for their usage. They stated that the DC_{16} op code would probably function as a compare index (CPX-9C) with direct memory address and FC16 as a compare index (CPX-BC) with extended memory address. Observation of the results of these instructions indicates that they probably do execute a compare index operation. The DC_{16} and FC_{16} op codes were changed to $9C_{16}$ and BC_{16} respectively because of the inability of Vendor A to explain the use of these bit patterns and because they are not specified op codes. It was later learned that Vendor A also elected to discontinue the use of these op codes.

The revised vectors (Functional Test Set II) are similar to the long and short test of Functional Test Set I in that there is no methodical sector by sector test of the microprocessor but rather an over-all test of the microprocessor as a single function. This, in itself, does not detract from the test but it does complicate analysis.

On a sector basis, (see Figure 3-1) the effectiveness of Test Set II is as follows:

1) I/O buffers and buses - meets test criteria.

2) A test of priority of $\overline{\text{HALT}}$ over $\overline{\text{NMI}}$ and $\overline{\text{IRQ}}$ was not performed.

3) Instruction Decode - Two instructions, AND A and BIT B, were not used. Many instructions were applied once but the results were not sensitized to the output. They are:

Mnemonic	Operation
COM A	Put 1's complement of Register A into Register A
LSR A	Logic shift right A
ASL A	Arithmetic shift A right
ROL A	Rotate A left, end around
TST A	Test A and set the flags
NEG B	Negate (2's complement) Register B
COM B	Put 1's complement of B into B
LSR B	Logic shift right B
ROR B	Rotate B right, end around
ASR B	Arithmetic shift B right
ASL B	Arithmetic shift B left
ROL B	Rotate B left, end around
DEC B	Decrement B
INC B	Increment B
SUB A	Subtract from A
SBC A	Subtract with borrow from A
BIT A	Compare bits of A with memory (AND)
SUB B	Subtract from B
ADD B	Add to B
ORA B	Logical OR with B
CMP B	Compare B to memory (subtract)
SBC B	Subtract with borrow from B
EOR B	Exclusive OR with B

4) Arithmetic and Logic Unit (ALU) - Many input pair combinations were not applied to the ALU in the various modes. Many more combinations were applied but not sensitized to the output.

5) Register Array - There is very little activity in the upper bits of the double length registers. Some tests were not performed in the lower bits and in the single registers.

Over-all Functional Test Set II is not so good as Test Set I because of these deficiencies. Many of the deficiencies in the Instruction Decode and the ALU sectors can be easily rectified by inserting a few op codes to sensitize the results of various operations to the outputs. Because of the ease of sensitizing states to outputs and the fact that Vendor A has added tests to catch known field failures in Test Set II, it was decided to modify Functional Test Set II and use it as a basis for the logic integrity tes' in the slash sheet. Additional vectors were specified and are described in the discussion and added to the test set. The resulting set of vectors satisfies the specified test criteria. Since this test, 1) verifies the logical operation of each sector, 2) tests all known interconnections, 3) verifies the instruction set, 4) includes tests for known sensitivities as supplied by the vendor; the resulting logic integrity and instruction integrity is estimated at 95 percent or greater.

Five sets of input timing and voltage conditions were specified which are considered worst cases of the dynamic portions of the uP. By applying the functional test set five times, using each of the input specifications, one obtains a high confidence of the integrity of the dynamic buses and dynamic timing circuitry under all input conditions.

3.3 Discussion

The block diagram (Figure 3-1) contained in Vendor A's data sheet shows the basic architecture of the Mc6800. The instruction decode and control section and the input/output parts are functional blocks which are required of the central processor in any computer. In a fixed-bit-length uP such as the Mc6800 (as opposed to a bit-slice uP such as the SBP0400) these sections are the boundaries of the uP chip, that is, the sections which interface with the rest of a system, and specify the operating mode of each internal section of the uP. The actual operation of these sections will be explained later in the report. The primary internal sections of the uP are the ALU (Arithmetic and Logic Unit), registers, internal data buses and instruction decode and control section. A person wishing to use this chip is initially interested in the following:

1) the width of the buses (number of bits)
2) the quantity and quality of operations that the ALU and associated circuitry can perform,
3) the quantity and size of the registers, and
4) the operations that can be performed on each register

Much of this information can be obtained from the block diagram (Figure 3-1), a list of op codes and their functions and the brief description contained in the Vendor A data sheet. However, for the purposes of test generation and test evaluation, this block diagram is lacking in much of the required detail. For example, the instruction decode and control is shown as a box with 16 inputs and three outputs. In actuality this block has additional inputs from the condition code register and probably hundreds of outputs controlling the registers, I/O ports, ALU and data buses. In order to evaluate the test vector set, an understanding is required of the function of this box and its interaction with the other sections. To aid in this understanding, expanded block diagrams were developed for this and other sections.

The supplied test vectors were presented as a table of "ones" and "zeros". If a logic diagram were made available, computer simulation programs could be used to evaluate their effectiveness. Since a logic diagram was not available, the test vector set effectiveness was determined by analyzing the functional requirements of each functional block of the Mc6800.

Since this method requires a description of the activity of each vector, a computer program, the "disassembler", was developed to convert the table of vectors to any assembly language equivalent.

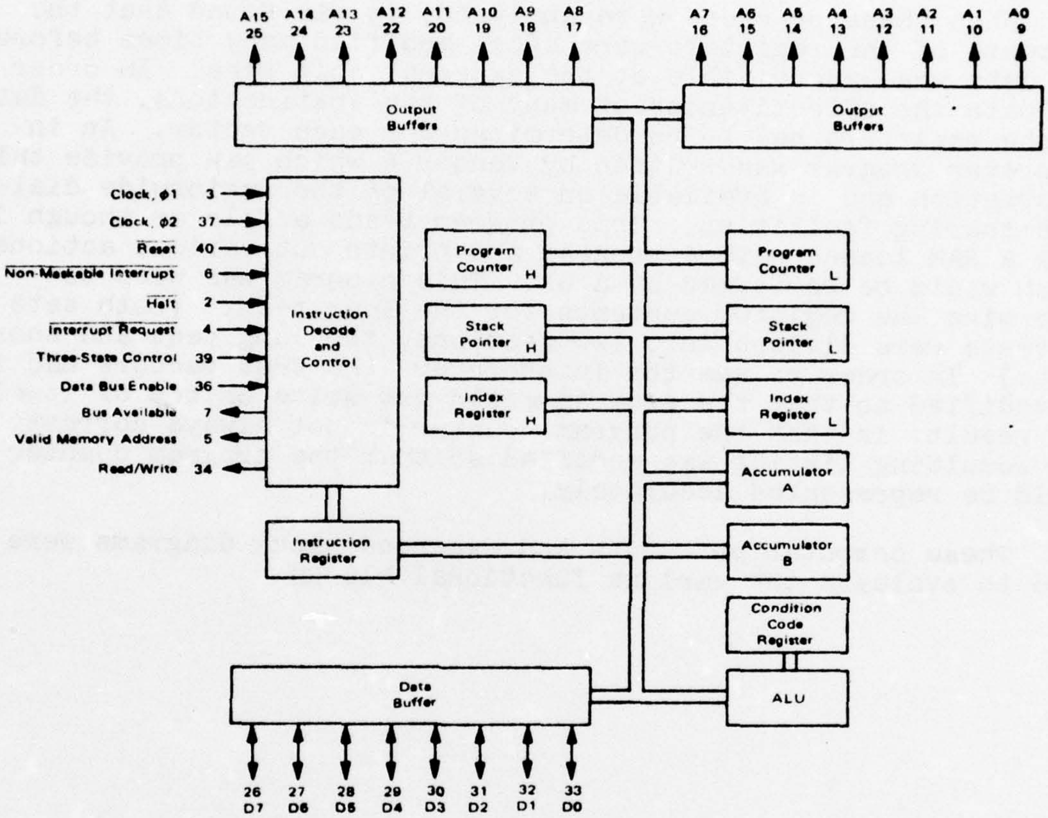


Figure 3-1. Mc6800 Block Diagram

If the uP were to read the test vectors from a uP-controlled RAM, the test would not be executed correctly. This is due to the fact that the program instructs the uP to store data into the memory locations where other instructions would be located. However, these vectors were never intended to be a program executed from RAM. They are instead an ordered set of vectors applied by an automatic tester.

When these op codes were analyzed, it was found that the contents of the registers were often modified many times before the data was made visible at the external chip pins. In order to evaluate the effectiveness of many of the instructions, the data in the registers had to be determined for each vector. An interpreter program was written by Vendor A which can provide this information and is available on several of the nationwide dial-up time-sharing facilities. This program reads a file as though it were a RAM loaded with a program and prints out various actions which would be performed by a uP. This program was used to determine the register contents for the long test. (Both sets of tests were divided into two sections, the long test and short test.) In order to use the Interpreter the test vectors had to be modified so that the program would not write on top of itself. The result is that the program counter is not always correct. The resulting listing was modified so that the program counter would be represented accurately.

These computer printouts and expanded block diagrams were used to evaluate the various functional blocks.

3.3.1 Data and Output, Buffers and Buses

The data and output buffer lines along with the read/write line interface the Mc6800 with memory (RAM and ROM) and input/output devices (teletype, line printer, floppy disc, modem, etc). The address lines select a particular I/O device or a particular memory address. If the read/write line is forced high by the uP, then the selected memory or input device will place data from the selected location onto the data bus. Once the data is stable (determined when the Phase 2 clock is brought high) the 6800 reads this data and puts it in the particular register specified by the instruction that it is executing. If the data is loaded into the instruction register then it is used as the next instruction. When the read/write line is brought low by the uP and RAM is addressed, then the uP will put data on the data bus. When the data is stable, it will be written into the selected location of RAM. Likewise, if an output device is specified such as the PIA in Figure 3-2 (peripheral interface adapter), then the data on the data bus is output to that device (e.g., to a printer or tape).

The circuitry analyzed in this section includes:

1) The data line buffers and associated chip pins,
2) An 8-bit data bus connecting data buffers, the registers and the ALU,
3) The address buffers and associated chip pins,
4) A 16-bit address bus from the program counter, stack pointer, index register and address incrementor/decrementor to the address buffers.

Each flip-flop in the data address buffers was checked for 1 to 1, 1 to 0, 0 to 0 and 0 to 1 transitions. In addition, each data bus line and the associated buffer flip-flops and chip pins were shown to be independent from each other. The buses interconnecting the internal registers with the output buffers are verified as an integral part of the internal registers.

Operation of the data and output buffers was verified. No additional test vectors are required.

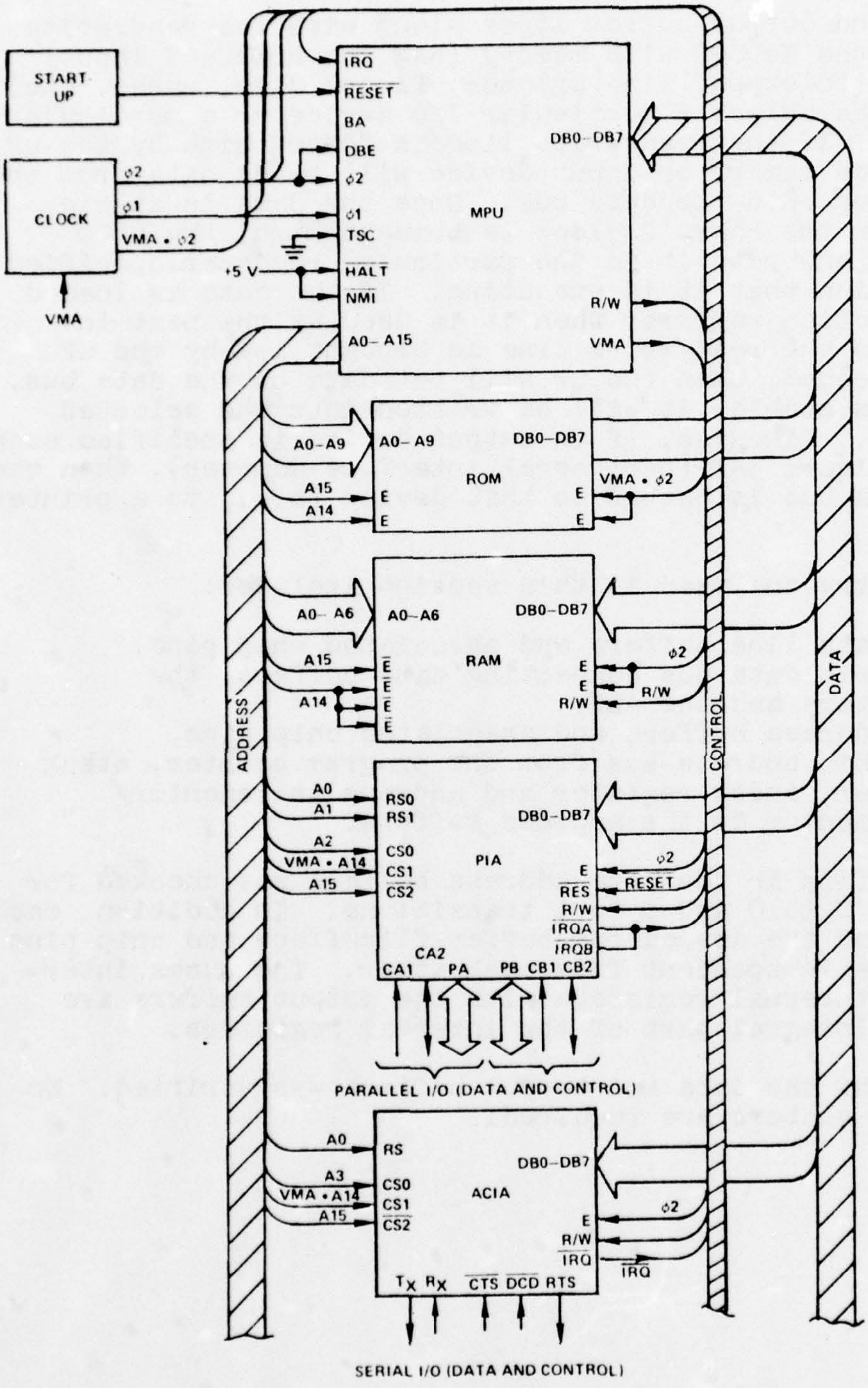


Figure 3-2. Typical Mc6800 System Wiring

3.3.2 Control

The first sector broken off as a separate entity from the Instruction Decode and Control Block was the basic timing and control section (Figure 3-3a). The inputs to this section, for example, interrupts, halt, etc., determine operation of the uP independent of the op codes which are being executed.

The clock inputs are inherent to all aspects of the microprocessor operation. The clock lines are automatically verified if the vector set is sufficient for each functional block. E.g., one of the clocks defines the time when a particular register is loaded with data. If the line if faulted, the register will either not be loaded or will be loaded even when loading is not specified. Both of these cases will be detected as bad data in the register.

The interrupt control lines ($\overline{\text{RESET}}$, $\overline{\text{HALT}}$, $\overline{\text{NONMASKABLE INTERRUPT}}$ ($\overline{\text{NMI}}$), and $\overline{\text{INTERRUPT REQUEST}}$ ($\overline{\text{IRQ}}$)), are used in situations where a low priority activity must be interrupted so that a higher priority activity can be executed. After the higher priority task is completed, the low priority task is resumed.

E.g. a uP could search through and modify a list contained in RAM concurrently with accepting data from a teletype and store it elsewhere in RAM. The search and modify task is a time-consuming task but one which can be interrupted and resumed without any loss of accuracy. The teletype task, on the other hand, is not time-consuming nor is it frequent relative to the uP's execution speed, but the task must be executed during the short period of time when the data is available. This task is easily handled using the interrupt capability. After each instruction of the table search is executed, the uP will check for a low on $\overline{\text{HALT}}$, $\overline{\text{NMI}}$ or $\overline{\text{IRQ}}$, and if any of these conditions are true, then the action indicated in Figure 3-3b is performed. By wiring the teletype to force NMI low when it is ready with data, the full power of the uP can be used on the table search until an interrupt is received by these uP. Then, the uP

1) stores the registers into the area of RAM which is addressed by the Stack Pointer Register,
2) loads the program counter with the contents of memory locations hex (hexadecimal) FFFC and FFFD, which contain the user defined starting location of the teletype servicing routine,
3) executes the teletype routine,
4) reloads the registers from the stack, and
5) continues with with search routine. Note that the Program Counter is stored into the stack at the beginning of the interrupt and restored at the end. Since the Program Counter always contains the address of the next instruction to be executed, the uP can resume the low priority task exactly where it left off.

The other interrupts and control are:

1) $\overline{\text{IRQ}}$ which, if the Mask bit in the Conditions Code Register is zero, affects the uP as described for $\overline{\text{NMI}}$ except that the Program Counter is loaded from memory locations hex FFF8 and FFF9.
2) $\overline{\text{RESET}}$ which causes the contents of memory locations hex FFFE and FFFF to be loaded into the Program Counter.
3) SWI (Software Interrupt), and software instruction, is similar to $\overline{\text{NMI}}$ except that locations hex FFFA and FFFB are used to load the Program Counter.
4) $\overline{\text{HALT}}$ which is not an interrupt in the sense of the other four but has a higher priority than $\overline{\text{NMI}}$, $\overline{\text{IRQ}}$, and SWI. Its function is to suspend operations of the uP.

A test of these functions should check that:

1) each performs the intended function.
2) The proper priority is maintained when any two or more are detected at the same time.
3) Each will perform independent of the previous instruction.

In the test set each of these control functions is exercised. Several combinations of these lines are activated together which verified the hierarchy of the priorities as follows:

1) RESET is recognized when RESET is applied with NMI and IRQ.
2) NMI is recognized when NMI is applied with IRQ but without RESET.

Figure 3-3b shows that an interrupt or halt occurs, independent of the instruction being executed. To guarantee that the test set verifies the independence of instructions, a gate-level logic diagram is required for verification. Based on available information, a satisfactory test is being performed on this sector.

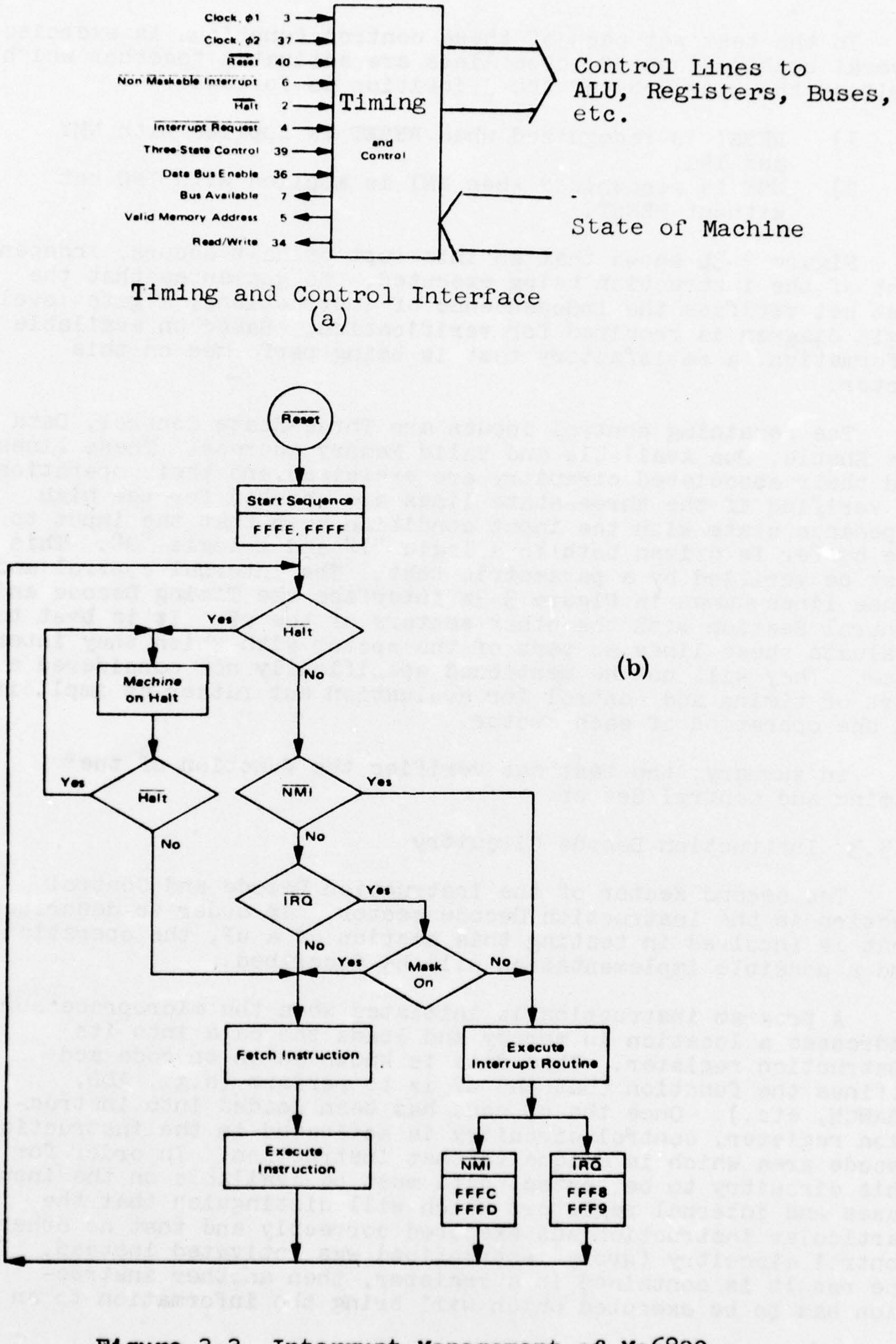
The remaining control inputs are Three-State Control, Data Bus Enable, Bus Available and Valid Memory address. These lines and their associated circuitry are exercised and their operation is verified if the three-state lines are checked for the high impedance state with the input condition such that the input to the buffer is driven both to a logic "1" and a logic "0". This must be verified by a parametric test. The internal control and sense lines shown in Figure 3-3a interface the Timing Decode and Control Section with the other sectors of the uP. It is best to evaluate these lines as part of the sector with which they interface. They will not be mentioned specifically nor considered a part of timing and control for evaluation but rather as implicit in the operation of each sector.

In summary, the test set verifies the function of the Timing and Control Sector.

3.3.3 Instruction Decode Circuitry

The second sector of the Instruction Decode and Control section is the Instruction Decode sector. In order to describe what is involved in testing this section of a uP, the operation and a possible implementation will be described.

A program instruction is initiated when the microprocessor addresses a location in memory and loads the data into its instruction register. This data is known as an op code and defines the function that the uP is to perform (e.g., ADD, BRANCH, etc.). Once the op code has been loaded into instruction register, control circuitry is activated in the instruction decode area which is unique to that instruction. In order for this circuitry to be tested, data must be available on the input buses and internal registers which will distinguish that the particular instruction was executed correctly and that no other control circuitry (wrong instruction) was activated instead. If the result is contained in a register, then another instruction has to be executed which will bring the information to an



Timing and Control Interface
(a)

Figure 3-3. Interrupt Management of Mc6800

externally observable point. Referring to Figure 1, the op code is brought in through the data buffer onto the data bus and then into the instruction register during the first clock cycle of each instruction. This eight-bit code is applied to the inputs of a decoder tree (Figure 3-4a), which in turn activates (e.g., pulls low) one of the outputs (instruction lines) of the decoder provided that the op code is a valid instruction. This instruction line activates a number of control lines which go directly to atomic functional areas of the uP. This is done by pulling the control line to the same logic level by means of diodes, shown as dots in the Programmed Logic Array (PLA) section of Figure 3-4a. For example, consider the ADA instruction which adds Accumulator A to Accumulator B and places the result in Accumulator A. The instruction line must activate the gates which:

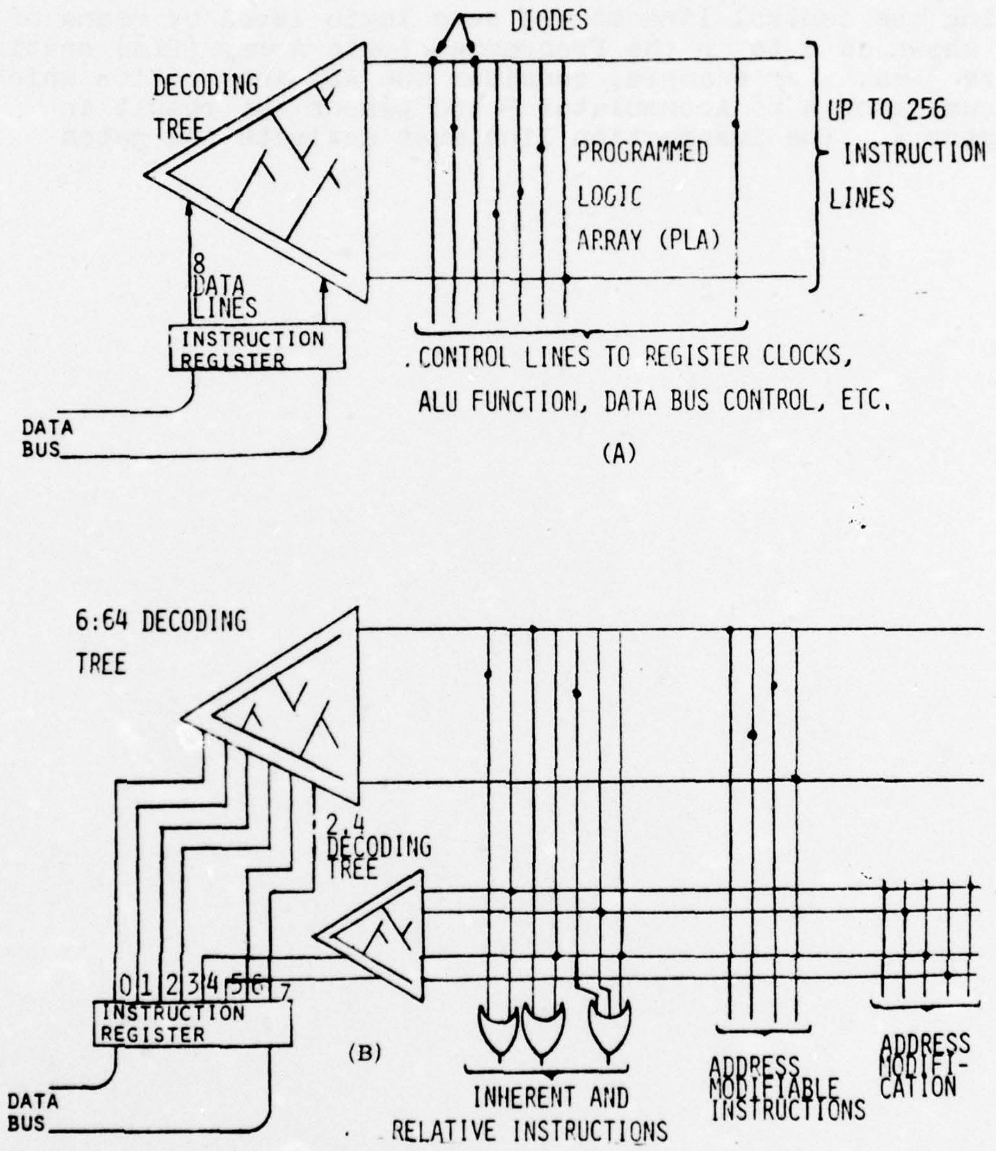


Figure 3-4. Microprocessor Instruction Decode Circuit

1) connect the accumulators through the ALU,
2) place the true/complement circuit in the true mode,
3) set the ALU in the "ADD" mode,
4) gate the appropriate bits of the condition code register,
5) specify the shifter in the "straight through" mode and
6) gate appropriate clocks to the output buffer and then back into A.

As a minimum test, all op codes must be used once. At first glance, it appears that the Mc6800 has 197 op codes. However, it is apparent that Vendor A's vector set does not use all op codes. In fact, only 123 of the 197 published op codes are used, plus two additional op codes which are not listed in their list of legal instructions. Since the logic diagram of the Mc6800 was not available, an alternate mechanization of the instruction decode section was devised in order to provide a basis for questions to Vendor A personnel. The resulting schematic is shown in Figure 3-4b. The decoding tree/PLA has been broken into two sections. For inherent and relative instructions, the net result is the same. All eight lines are required to define an instruction and all of these instructions must be used for a complete test.

Since the address modification circuitry and its control are common to all address modifiable instructions there are only 106 unique instructions. The unique instructions are the 66 instructions from 00 through 5F which can be identified as a group by the three higher order bits, and the 40 modifiable instructions from 60 through FF. For example, op codes 80, 90, A0 and B0 produce the same internal operation (subtract Memory from Register A) except in the address determination circuitry. A test of the instruction decode circuitry requires the verification of the 66 inherent and relative instructions, 40 address-modifiable instructions and the 4 addressing modes.

When the mechanization Figure 3-4b was presented to Vendor A personnel, they stated that the actual mechanization is based on this idea and that the op codes used are sufficient to exercise the instruction decode section.

Vendor A personnel were also asked to explain their use of op codes 00 and 03 which do not appear in their list of instructions. They stated that there are four no-op's which are : 00, 01, 02 and 03. However 00, 02 and 03 may under certain conditions cause unexpected operations. Therefore, 01 is the only recommended NOP for general programming use. Vendor A used 00 and 03 because certain testable areas, unspecified to us, could be tested in fewer vectors. They stated that these same areas could be tested using their listed instructions, however, it would require more test vectors. Since Vendor A did not specify which areas were covered using the no-op's and without a logic diagram, it is not possible to define the tests for these areas. This is the intent of Built In Test Electronics (BITE), and it appears that Vendor A is taking advantage of this type of circuitry. Since the existence of this circuitry is not published and a logic diagram is not available, the effects of these op codes cannot be evaluated nor could anyone else take advantage of their existence as BITE. Other than these. Vendor A stated that no other unlisted op codes exist.

Using the above criteria (Figure 3-4b) for unique operation codes, it was found that AND A (AND Register A with memory), BIT B (bit test Register B with memory), CPX (compare Index Register), and the normal NOP (OP code = 01) were the only op codes which were never used. However, there were several instances where the results of an op code or a group of op codes were not sensitized to the outputs. In the final test set for the slash sheet, op codes were inserted, as necessary, to sensitize the results of these op codes. These could have been verified by adding vectors to the end of the test, however, this would have lengthened the test set significantly. Vectors were added to the end of the Vendor A's test set to detect faults not previously checked. Changes made for the recommended vector set are as follows:

--At Vector 40 a TXS (transfer the contents - 1 of the Index Register to the Stack Pointer Register) was executed which wrote over the results of the INS (increment the Stack Pointer Register) at Vector 40. Since INS is not used elsewhere in the test, the integrity of this instruction is never verified. Verification of INS was performed by dumping the contents of the Stack Pointer before the TSX instruction.

--At Vector 133 a CLRA (clear Register A) and at 139 a TAB (transfer Register A to Register B) were executed. These two instructions destroy the results of COMB (4's complement the content of Register B), ASLB (arithmetic shift left Register B), and TBA (transfer Register B to Register A). COMB and ASLB were not used elsewhere. Due to the sequence of these instructions, the results of all three instructions are contained in Register A and were sensitized to the output by examing the contents of Register A before the CLRA.

--At Vector 359 the results of LDX (load the Index Register) with 8000 Hex were not examined. This op code is used elsewhere but only with the new data = 0000 Hex and with the previous state of the register unknown. Therefore the operation of LDX is not guaranteed and, furthermore, the operation of the Index Register is, for the most part, not verified. The faults detected by the Vendor A test and the vectors added are described in more detail in the section on the Register Array.

--At Vector 586 a CLRB (clear Register B) destroys the results of several instructions preceeding it. These instructions were not used elsewhere and, therefore, their operation was not verified:

VECTOR	MNEMONIC	OPERATION
578	RORB	Rotate right Register B
580	LSRB	Logical shift right Register B
582	ASRB	Arithmetic shift right Register B
584	ROLB	Rotate left Register B

This series of instructions is a case where a group of similar instructions are concatenated and where the intermediate results (contained in Register B) are not examined. It is possible for a fault in the shift circuitry to be missed when one failure masks another from a previous shift. For this case, it is recommended that only the final result be examined after the ROLB since additional shift instructions are required as detailed in the ALU section of the report. It is not probable that a fault would be masked here and not detected elsewhere. At Vectors 594 and 596 the only occurences of DECB (decrement Register B) and INCB (increment Register B) are executed back to back without examining the contents of Register B. These instructions are very similar in nature and may use common circuitry which would mask a failure by either instruction. Therefore a STAB (store Accumulator B in memory) instruction was inserted between them.

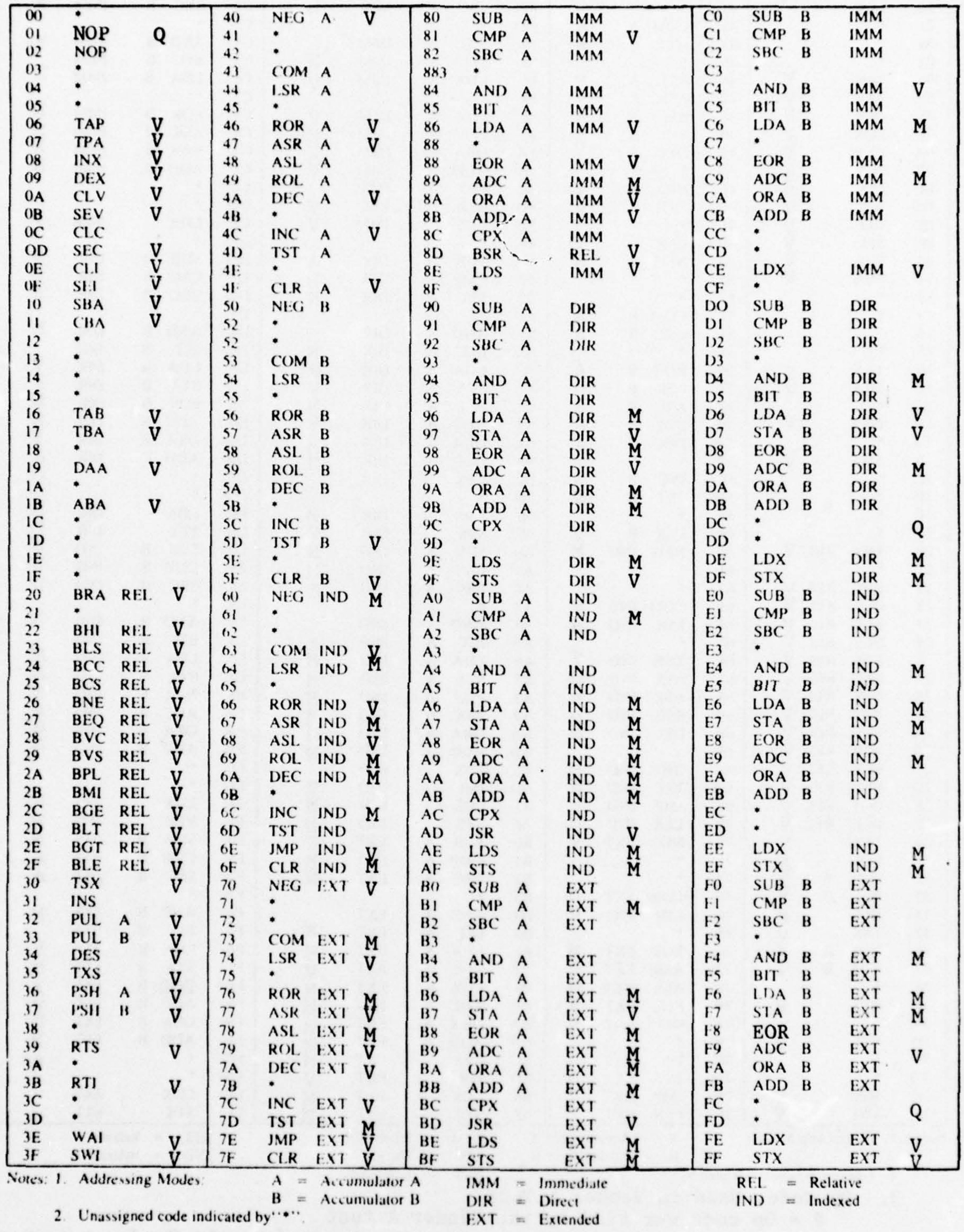
--In Vector 630, a CMPB (compare Register B with memory) is used but its operation is not checked. There is a BLT (branch on less than) immediately following the CMPB which could check its operation. However, the address is the same for either "less than" or "not less than". Therefore, it is impossible to determine if the branch is taken. A branch instruction was inserted which detects the conditions of the Condition Code Register relative to the CMPB instruction. The existing branch was left unchanged in case the test designer had a particular reason for the instruction. Since this type of branch (same branch address for either condition) was used both in the original and the revised tests, there is reason to believe that a fault is being checked. The CMPB was added again to insure that the Condition Code Register is changed by the instruction.

--At Vector 661 an RTI (return from interrupt) overwrites the A, B and Condition Code Registers which contain the results of the following instructions:

VECTOR	MNEMONIC	OPERATION
576	TP A	Transfer flags to accum A
586	CLR B	Clear accum B
588	TST B	Set flags according to accum B
594	DEC B	Decrement accum B
596	INC B	Increment B
598	NEG B	Negate (2's complement) B
600	COM A	1's complement A
602	ASL A	Arithmetic shift left A
604	ORA B	OR to B
607	EQR B	Exclusive OR to B
617	ADD B	Add to B
620	SUB B	Subtract from B
626	CMP B	Compare to B
634	SUB A	Subtract from A
637	BIT A	Bit test with A
646	SBC A	Subtract with borrow from A
655	SBC B	Subtract with borrow from B

Except for the first three, none of the above op codes were verified elsewhere in the test. Several op codes were inserted to obtain the desired visibility.

Figure 3-5 lists the Mc6800 instruction set and identifies which instructions were verified in Vendor A's test. Figure 3-6 is the same as Figure 3-5 with the addition of op codes which the Vendor A test applies but does not verify. Figure 3-7 identifies the op codes which were used and verified in the final test set. The final test verifies all unique instructions. Those instructions not shown in Figure 3-7 as being exercised are in fact checked. This is due to the design of the instruction decode circuitry as previously described.



00	•			40	NEG	A	V	80	SUB	A	IMM		C0	SUB	B	IMM	
01	NOP		Q	41	•			81	CMP	A	IMM	V	C1	CMP	B	IMM	
02	NOP			42	•			82	SBC	A	IMM		C2	SBC	B	IMM	
03	•			43	COM	A		883	•				C3	•			
04	•			44	LSR	A		84	AND	A	IMM		C4	AND	B	IMM	V
05	•			45	•			85	BIT	A	IMM		C5	BIT	B	IMM	
06	TAP		V	46	ROR	A	V	86	LDA	A	IMM	V	C6	LDA	B	IMM	M
07	TPA		V	47	ASR	A	V	88	•				C7	•			
08	INX		V	48	ASL	A		88	EOR	A	IMM	V	C8	EOR	B	IMM	
09	DEX		V	49	ROL	A		89	ADC	A	IMM	M	C9	ADC	B	IMM	M
0A	CLV		V	4A	DEC	A	V	8A	ORA	A	IMM	V	CA	ORA	B	IMM	
0B	SEV		V	4B	•			8B	ADD	A	IMM	V	CB	ADD	B	IMM	
0C	CLC			4C	INC	A	V	8C	CPX	A	IMM		CC	•			
OD	SEC		V	4D	TST	A		8D	BSR		REL	V	CD	•			
0E	CLI		V	4E	•			8E	LDS		IMM	V	CE	LDX		IMM	V
0F	SEI		V	4F	CLR	A	V	8F	•				CF	•			
10	SBA		V	50	NEG	B		90	SUB	A	DIR		DO	SUB	B	DIR	
11	CBA		V	52	•			91	CMP	A	DIR		D1	CMP	B	DIR	
12	•			52	•			92	SBC	A	DIR		D2	SBC	B	DIR	
13	•			53	COM	B		93	•				D3	•			
14	•			54	LSR	B		94	AND	A	DIR		D4	AND	B	DIR	M
15	•			55	•			95	BIT	A	DIR		D5	BIT	B	DIR	
16	TAB		V	56	ROR	B		96	LDA	A	DIR	M	D6	LDA	B	DIR	V
17	TBA		V	57	ASR	B		97	STA	A	DIR	V	D7	STA	B	DIR	V
18	•			58	ASL	B		98	EOR	A	DIR	M	D8	EOR	B	DIR	
19	DAA		V	59	ROL	B		99	ADC	A	DIR	V	D9	ADC	B	DIR	M
1A	•			5A	DEC	B		9A	ORA	A	DIR	M	DA	ORA	B	DIR	
1B	ABA		V	5B	•			9B	ADD	A	DIR	M	DB	ADD	B	DIR	
1C	•			5C	INC	B		9C	CPX		DIR		DC	•			Q
1D	•			5D	TST	B	V	9D	•				DD	•			
1E	•			5E	•			9E	LDS		DIR	M	DE	LDX		DIR	M
1F	•			5F	CLR	B	V	9F	STS		DIR	V	DF	STX		DIR	M
20	BRA	REL	V	60	NEG	IND	M	A0	SUB	A	IND		E0	SUB	B	IND	
21	•			61	•			A1	CMP	A	IND	M	E1	CMP	B	IND	
22	BHI	REL	V	62	•			A2	SBC	A	IND		E2	SBC	B	IND	
23	BLS	REL	V	63	COM	IND	V	A3	•				E3	•			
24	BCC	REL	V	64	LSR	IND	M	A4	AND	A	IND		E4	AND	B	IND	M
25	BCS	REL	V	65	•			A5	BIT	A	IND		E5	BIT	B	IND	
26	BNE	REL	V	66	ROR	IND	V	A6	LDA	A	IND	M	E6	LDA	B	IND	M
27	BEQ	REL	V	67	ASR	IND	M	A7	STA	A	IND	M	E7	STA	B	IND	M
28	BVC	REL	V	68	ASL	IND	V	A8	EOR	A	IND	M	E8	EOR	B	IND	
29	BVS	REL	V	69	ROL	IND	M	A9	ADC	A	IND	M	E9	ADC	B	IND	M
2A	BPL	REL	V	6A	DEC	IND	M	AA	ORA	A	IND	M	EA	ORA	B	IND	
2B	BMI	REL	V	6B	•			AB	ADD	A	IND	M	EB	ADD	B	IND	
2C	BGE	REL	V	6C	INC	IND	M	AC	CPX		IND		EC	•			
2D	BLT	REL	V	6D	TST	IND		AD	JSR		IND	V	ED	•			
2E	BGT	REL	V	6E	JMP	IND	V	AE	LDS		IND	M	EE	LDX		IND	M
2F	BLE	REL	V	6F	CLR	IND	M	AF	STS		IND	M	EF	STX		IND	M
30	TSX		V	70	NEG	EXT	V	B0	SUB	A	EXT		F0	SUB	B	EXT	
31	INS			71	•			B1	CMP	A	EXT	M	F1	CMP	B	EXT	
32	PUL	A	V	72	•			B2	SBC	A	EXT		F2	SBC	B	EXT	
33	PUL	B	V	73	COM	EXT	M	B3	•				F3	•			
34	DES		V	74	LSR	EXT	V	B4	AND	A	EXT		F4	AND	B	EXT	M
35	TXS		V	75	•			B5	BIT	A	EXT		F5	BIT	B	EXT	
36	PSH	A	V	76	ROR	EXT	M	B6	LDA	A	EXT	M	F6	LDA	B	EXT	M
37	PSH	B	V	77	ASR	EXT	V	B7	STA	A	EXT	V	F7	STA	B	EXT	M
38	•			78	ASL	EXT	M	B8	EOR	A	EXT	M	F8	EOR	B	EXT	
39	RTS		V	79	ROL	EXT	V	B9	ADC	A	EXT	M	F9	ADC	B	EXT	V
3A	•			7A	DEC	EXT	V	BA	ORA	A	EXT	M	FA	ORA	B	EXT	
3B	RTI		V	7B	•			BB	ADD	A	EXT	M	FB	ADD	B	EXT	
3C	•			7C	INC	EXT	V	BC	CPX		EXT		FC	•			Q
3D	•			7D	TST	EXT	M	BD	JSR		EXT	V	FD	•			
3E	WAI		V	7E	JMP	EXT	V	BE	LDS		EXT	M	FE	LDX		EXT	V
3F	SWI		V	7F	CLR	EXT	V	BF	STS		EXT	M	FF	STX		EXT	V

Notes: 1. Addressing Modes:

A	=	Accumulator A	IMM	=	Immediate	REL	=	Relative
B	=	Accumulator B	DIR	=	Direct	IND	=	Indexed
			EXT	=	Extended			

2. Unassigned code indicated by "•".

3. Op Codes Used in Vendor A Test
   V = Op code verified by the Vendor A test
   M = Address modifications of Op code verified by the Vendor A test
   Q = Op code applied but not in Vendor A's list of legal Op codes
   Blank = Unique instructions which were not used.

Figure 3-5. Verified Instruction Set

00	•			40	NEG	A	V	80	SUB	A	IMM	N	C0	SUB	B	IMM	A
01	NOP		Q	41	•			81	CMP	A	IMM	V	C1	CMP	B	IMM	N
02	NOP			42	•			82	SBC	A	IMM	N	C2	SBC	B	IMM	N
03	•			43	COM	A	A	883	•				C3	•			
04	•			44	LSR	A	A	84	AND	A	IMM		C4	AND	B	IMM	V
05	•			45	•			85	BIT	A	IMM	N	C5	BIT	B	IMM	
06	TAP		V	46	ROR	A	V	86	LDA	A	IMM	V	C6	LDA	B	IMM	M
07	TPA		V	47	ASR	A	V	88	•				C7	•			
08	INX		V	48	ASL	A	A	88	EOR	A	IMM	V	C8	EOR	B	IMM	N
09	DEX		V	49	ROL	A	A	89	ADC	A	IMM	M	C9	ADC	B	IMM	M
0A	CLV		V	4A	DEC	A	V	8A	ORA	A	IMM	V	CA	ORA	B	IMM	N
0B	SEV		V	4B	•			8B	ADD	A	IMM	V	CB	ADD	B	IMM	N
0C	CLC		A	4C	INC	A	V	8C	CPX	A	IMM		CC	•			
OD	SEC		V	4D	TST	A	A	8D	BSR		REL	V	CD	•			
0E	CLI		V	4E	•			8E	LDS		IMM	V	CE	LDX		IMM	V
0F	SEI		V	4F	CLR	A	V	8F	•				CF	•			
10	SBA		V	50	NEG	B	A	90	SUB	A	DIR	A	DO	SUB	B	DIR	N
11	CBA		V	52	•			91	CMP	A	DIR	M	D1	CMP	B	DIR	N
12	•			52	•			92	SBC	A	DIR	N	D2	SBC	B	DIR	N
13	•			53	COM	B	A	93	•				D3	•			
14	•			54	LSR	B	A	94	AND	A	DIR		D4	AND	B	DIR	M
15	•			55	•			95	BIT	A	DIR	N	D5	BIT	B	DIR	
16	TAB		V	56	ROR	B	A	96	LDA	A	DIR	M	D6	LDA	B	DIR	V
17	TBA		V	57	ASR	B	A	97	STA	A	DIR	V	D7	STA	B	DIR	V
18	•			58	ASL	B	A	98	EOR	A	DIR	M	D8	EOR	B	DIR	N
19	DAA		V	59	ROL	B	A	99	ADC	A	DIR	V	D9	ADC	B	DIR	M
1A	•			5A	DEC	B	A	9A	ORA	A	DIR	M	DA	ORA	B	DIR	N
1B	ABA		V	5B	•			9B	ADD	A	DIR	M	DB	ADD	B	DIR	A
1C	•			5C	INC	B	A	9C	CPX		DIR		DC	•			Q
1D	•			5D	TST	B	V	9D	•				DD	•			
1E	•			5E	•			9E	LDS		DIR	M	DE	LDX		DIR	M
1F	•			5F	CLR	B	V	9F	STS		DIR	V	DF	STX		DIR	M
20	BRA	REL	V	60	NEG	IND	M	A0	SUB	A	IND	N	E0	SUB	B	IND	N
21	•			61	•			A1	CMP	A	IND	M	E1	CMP	B	IND	N
22	BHI	REL	V	62	•			A2	SBC	A	IND	A	E2	SBC	B	IND	N
23	BLS	REL	V	63	COM	IND	V	A3	•				E3	•			
24	BCC	REL	V	64	LSR	IND	M	A4	AND	A	IND		E4	AND	B	IND	M
25	BCS	REL	V	65	•			A5	BIT	A	IND	A	E5	BIT	B	IND	
26	BNE	REL	V	66	ROR	IND	V	A6	LDA	A	IND	M	E6	LDA	B	IND	M
27	BEQ	REL	V	67	ASR	IND	M	A7	STA	A	IND	M	E7	STA	B	IND	M
28	BVC	REL	V	68	ASL	IND	V	A8	EOR	A	IND	M	E8	EOR	B	IND	N
29	BVS	REL	V	69	ROL	IND	M	A9	ADC	A	IND	M	E9	ADC	B	IND	M
2A	BPL	REL	V	6A	DEC	IND	M	AA	ORA	A	IND	M	EA	ORA	B	IND	A
2B	BMI	REL	V	6B	•			AB	ADD	A	IND	M	EB	ADD	B	IND	N
2C	BGE	REL	V	6C	INC	IND	M	AC	CPX		IND		EC	•			
2D	BLT	REL	V	6D	TST	IND	N	AD	JSR		IND	V	ED	•			
2E	BGT	REL	V	6E	JMP	IND	V	AE	LDS		IND	M	EE	LDX		IND	M
2F	BLE	REL	V	6F	CLR	IND	M	AF	STS		IND	M	EF	STX		IND	M
30	TSX		V	70	NEG	EXT	V	B0	SUB	A	EXT	N	F0	SUB	B	EXT	N
31	INS		A	71	•			B1	CMP	A	EXT	M	F1	CMP	B	EXT	A
32	PUL	A	V	72	•			B2	SBC	A	EXT	N	F2	SBC	B	EXT	A
33	PUL	B	V	73	COM	EXT	M	B3	•				F3	•			
34	DES		V	74	LSR	EXT	V	B4	AND	A	EXT		F4	AND	B	EXT	M
35	TXS		V	75	•			B5	BIT	A	EXT	N	F5	BIT	B	EXT	
36	PSH	A	V	76	ROR	EXT	M	B6	LDA	A	EXT	M	F6	LDA	B	EXT	M
37	PSH	B	V	77	ASR	EXT	V	B7	STA	A	EXT	V	F7	STA	B	EXT	M
38	•			78	ASL	EXT	M	B8	EOR	A	EXT	M	F8	EOR	B	EXT	A
39	RTS		V	79	ROL	EXT	V	B9	ADC	A	EXT	M	F9	ADC	B	EXT	V
3A	•			7A	DEC	EXT	V	BA	ORA	A	EXT	M	FA	ORA	B	EXT	N
3B	RTI		V	7B	•			BB	ADD	A	EXT	M	FB	ADD	B	EXT	N
3C	•			7C	INC	EXT	V	BC	CPX		EXT	M	FC	•			Q
3D	•			7D	TST	EXT	M	BD	JSR		EXT	V	FD	•			
3E	WAI		V	7E	JMP	EXT	V	BE	LDS		EXT	M	FE	LDX		EXT	V
3F	SWI		V	7F	CLR	EXT	V	BF	STS		EXT	M	FF	STX		EXT	V

Notes: 1. Addressing Modes:

A = Accumulator A	IMM = Immediate	REL = Relative
B = Accumulator B	DIR = Direct	IND = Indexed
	EXT = Extended	

2. Unassigned code indicated by "•"

3. Op Codes Used in Vendor A Test

V = Op code verified by the Vendor A test

M = Address modifications of Op code verified by the Vendor A test

A = Op codes applied but not verified

N = Address modifications of Op codes applied but not verified

Q = Op code applied but not in Vendor A's list of legal Op codes

Blank = Unique instructions which were not used.

Figure 3-6. Verified Instruction Set With Op Code

00	•			40	NEG	A	U	80	SUB	A	IMM		C0	SUB	B	IMM	U
01	NOP		U	41	•			81	CMP	A	IMM	U	C1	CMP	B	IMM	
02	NOP		U	42	•			82	SBC	A	IMM	U	C2	SBC	B	IMM	U
03	•			43	COM	A	U	883	•				C3	•			
04	•			44	LSR	A	U	84	AND	A	IMM		C4	AND	B	IMM	U
05	•			45	•			85	BIT	A	IMM		C5	BIT	B	IMM	
06	TAP		U	46	ROR	A	U	86	LDA	A	IMM	U	C6	LDA	B	IMM	U
07	TPA		U	47	ASR	A	U	88	•				C7	•			
08	INX		U	48	ASL	A	U	88	EOR	A	IMM	U	C8	EOR	B	IMM	
09	DEX		U	49	ROL	A	U	89	ADC	A	IMM	U	C9	ADC	B	IMM	
0A	CLV		U	4A	DEC	A	U	8A	ORA	A	IMM	U	CA	ORA	B	IMM	U
0B	SEV		U	4B	•			8B	ADD	A	IMM	U	CB	ADD	B	IMM	
0C	CLC		U	4C	INC	A	U	8C	CPX	A	IMM		CC	•			
OD	SEC		U	4D	TST	A	U	8D	BSR		REL	U	CD	•			
0E	CLI		U	4E	•			8E	LDS		IMM	U	CE	LDX		IMM	U
0F	SEI		U	4F	CLR	A	U	8F	•				CF	•			
10	SBA		U	50	NEG	B	U	90	SUB	A	DIR	U	DO	SUB	B	DIR	
11	CBA		U	52	•			91	CMP	A	DIR		D1	CMP	B	DIR	
12	•			52	•			92	SBC	A	DIR	U	D2	SBC	B	DIR	
13	•			53	COM	B	U	93	•				D3	•			
14	•			54	LSR	B	U	94	AND	A	DIR		D4	AND	B	DIR	
15	•			55	•			95	BIT	A	DIR		D5	BIT	B	DIR	
16	TAB		U	56	ROR	B	U	96	LDA	A	DIR	U	D6	LDA	B	DIR	U
17	TBA		U	57	ASR	B	U	97	STA	A	DIR	U	D7	STA	B	DIR	U
18	•			58	ASL	B	U	98	EOR	A	DIR		D8	EOR	B	DIR	U
19	DAA		U	59	ROL	B	U	99	ADC	A	DIR	U	D9	ADC	B	DIR	
1A	•			5A	DEC	B	U	9A	ORA	A	DIR		DA	ORA	B	DIR	U
1B	ABA		U	5B	•			9B	ADD	A	DIR		DB	ADD	B	DIR	U
1C	•			5C	INC	B	U	9C	CPX		DIR	U	DC	•			
1D	•			5D	TST	B	U	9D	•				DD	•			
1E	•			5E	•			9E	LDS		DIR	U	DE	LDX		DIR	U
1F	•			5F	CLR	B	U	9F	STS		DIR	U	DF	STX		DIR	
20	BRA	REL	U	60	NEG	IND		A0	SUB	A	IND		E0	SUB	B	IND	
21	•			61	•			A1	CMP	A	IND		E1	CMP	B	IND	
22	BHI	REL	U	62	•			A2	SBC	A	IND	U	E2	SBC	B	IND	
23	BLS	REL	U	63	COM	IND	U	A3	•				E3	•			
24	BCC	REL	U	64	LSR	IND		A4	AND	A	IND		E4	AND	B	IND	
25	BCS	REL	U	65	•			A5	BIT	A	IND	U	E5	BIT	B	IND	
26	BNE	REL	U	66	ROR	IND	U	A6	LDA	A	IND	U	E6	LDA	B	IND	U
27	BEQ	REL	U	67	ASR	IND		A7	STA	A	IND	U	E7	STA	B	IND	U
28	BVC	REL	U	68	ASL	IND	U	A8	EOR	A	IND	U	E8	EOR	B	IND	U
29	BVS	REL	U	69	ROL	IND		A9	ADC	A	IND	U	E9	ADC	B	IND	
2A	BPL	REL	U	6A	DEC	IND		AA	ORA	A	IND		EA	ORA	B	IND	U
2B	BMI	REL	U	6B	•			AB	ADD	A	IND		EB	ADD	B	IND	
2C	BGE	REL	U	6C	INC	IND		AC	CPX		IND		EC	•			
2D	BLT	REL	U	6D	TST	IND	U	AD	JSR		IND	U	ED	•			
2E	BGT	REL	U	6E	JMP	IND	U	AE	LDS		IND	U	EE	LDX		IND	U
2F	BLE	REL	U	6F	CLR	IND		AF	STS		IND		EF	STX		IND	
30	TSX		U	70	NEG	EXT	U	B0	SUB	A	EXT		F0	SUB	B	EXT	
31	INS		U	71	•			B1	CMP	A	EXT		F1	CMP	B	EXT	U
32	PUL	A	U	72	•			B2	SBC	A	EXT		F2	SBC	B	EXT	
33	PUL	B	U	73	COM	EXT		B3	•				F3	•			
34	DES		U	74	LSR	EXT	U	B4	AND	A	EXT	U	F4	AND	B	EXT	
35	TXS		U	75	•			B5	BIT	A	EXT		F5	BIT	B	EXT	
36	PSH	A	U	76	ROR	EXT		B6	LDA	A	EXT		F6	LDA	B	EXT	U
37	PSH	B	U	77	ASR	EXT	U	B7	STA	A	EXT	U	F7	STA	B	EXT	
38	•			78	ASL	EXT		B8	EOR	A	EXT		F8	EOR	B	EXT	
39	RTS		U	79	ROL	EXT	U	B9	ADC	A	EXT	U	F9	ADC	B	EXT	U
3A	•			7A	DEC	EXT	U	BA	ORA	A	EXT		FA	ORA	B	EXT	
3B	RTI		U	7B	•			BB	ADD	A	EXT		FB	ADD	B	EXT	
3C	•			7C	INC	EXT	U	BC	CPX		EXT	U	FC	•			
3D	•			7D	TST	EXT		BD	JSR		EXT	U	FD	•			
3E	WAI		U	7E	JMP	EXT	U	BE	LDS		EXT		FE	LDX		EXT	U
3F	SWI		U	7F	CLR	EXT	U	BF	STS		EXT	U	FF	STX		EXT	U

Notes: 1. Addressing Modes: A = Accumulator A; B = Accumulator B; IMM = Immediate; DIR = Direct; EXT = Extended; REL = Relative; IND = Indexed

2. Unassigned code indicated by "•"

3. Op Codes Used in Final Test Indicated by "U"

Figure 3-7. Verified Op Codes

3.3.4 Arithmetic Logic Unit

While evaluating the effectiveness of the vectors on the ALU sector, it was determined that the ALU sector should be further subdivided. Based on these observations and conversations with Vendor A personnel, the diagram in Figure 3-8 was developed for evaluation. The significant factor in this diagram is that the partition labeled ALU only performs four functions: ADD, AND, OR and EXOR. The remaining arithmetic and logic functions are implemented with a ones complementor (eight-bit EXOR) on the input lines and a shifter on the output lines. This is significantly simpler than a eight-bit 54181 and therefore, more straightforward with respect to evaluation and generation of tests. In addition, it was observed that the ALU could operate on both accumulators and return the result to Accumulator A in two cycles. Since the second cycle is required to load Accumulator A with the result, only one cycle is available for transfer of the data from each accumulator to the appropriate part of the ALU. This indicates that two separate data buses are used in this operation and are indicated in Figure 3-8 as Buses c and f. Using similar reasoning on memory to with accumulator operations, Buses b and d were added.

The test philosophy normally applied to an arithmetic adder/subtractor whose mechanization is not known is to apply all possible input combinations to each bit. That is, for each mode, add and subtract:

1) Apply all possible inputs (0 & 0, 0 & 1, 1 & 0 and 1 & 1) to each adder input pair with its carry-in a "zero".
2) Apply all possible inputs to each adder input pair with its carry-in a "one".

In the Mc6800, subtraction is effected by complementing the subtrahend and adding with carry-in equal to "one", that is, converting the subtrahend to a two's complement number and adding. The two's complement of a number is generated by inverting each bit of a number and adding one.

Examples:

```
Sign                      LSB
   0  0  0  0  1  1  0  1  =  +13
   1  1  1  1  0  0  1  1  =  -13
```

For an eight-bit number the maximum signed numbers are:

```
Sign
   0  1  1  1  1  1  1  1  =  +127
   1  0  0  0  0  0  0  0  =  -128
```

The sign of a number is determined by the state of B7, "zero" = positive and "one" = negative. The remaining bits indicate the magnitude for a positive number, and the two's complement representation of the magnitude for a negative number. The necessary tasks are therefore to:

1) verify the eight possible inputs to each bit of the adder,
2) verify that the complementing circuitry will complement both a "one" and a "zero" for each bit,
3) check bit independence and
4) verify decimal adjust circuitry.

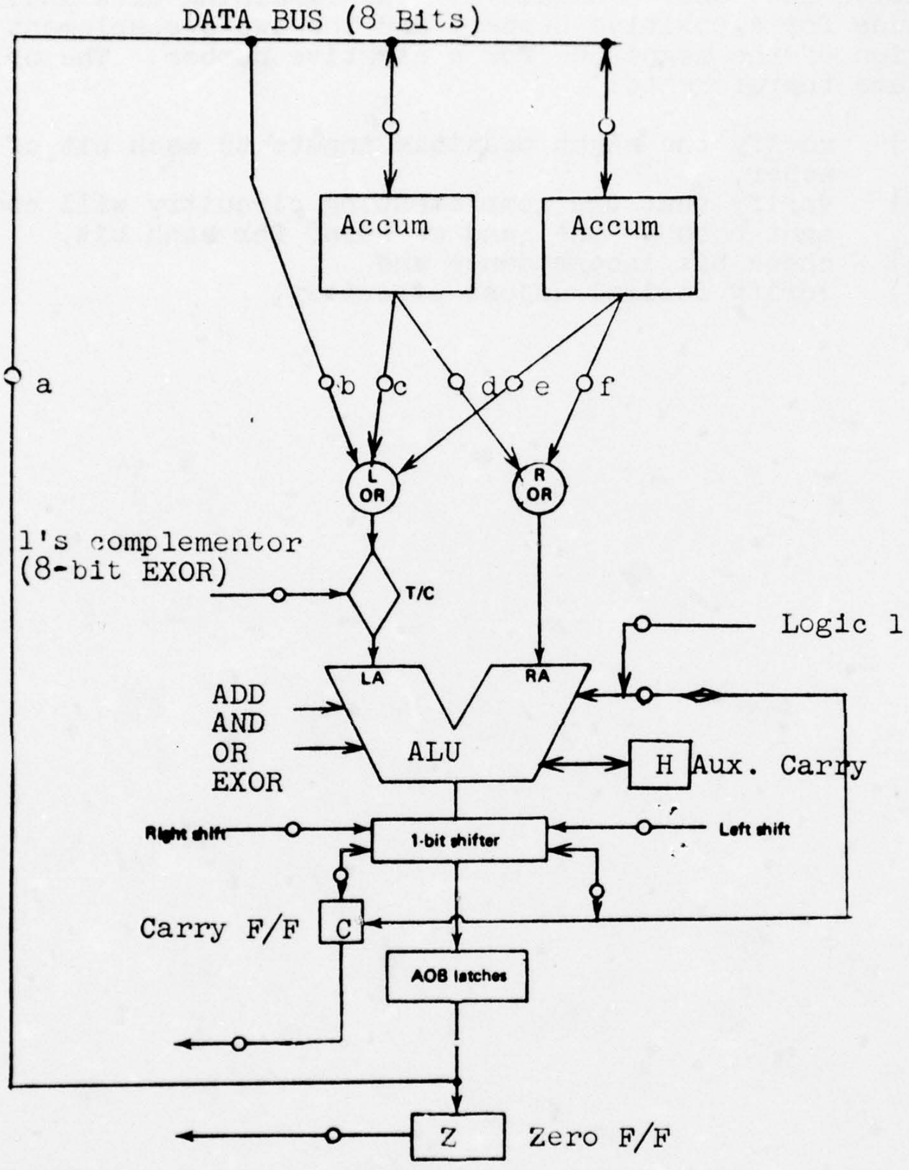


Figure 3-8. Sub-sectoring for ALU and Condition Code Register

The test philosophy normally applied to the ALU for logic operations is to:

1) Apply the following input conditions to each input pair:

0 & 0, 0 & 1, 1 & 0 and 1 & 1	during EXOR operations
0 & 0, 0 & 1, and 1 & 0	during OR operations
0 & 1, 1 & 0 and 1 & 1	during AND operations

2) Check that for shift left and for shift right operations both "0" and "1" are shifted from each bit into a "0" and a "1" in each adjoining bit. Therefore, four shift left and four shift right combinations are recommended for each bit.

Functional Test Set II is much less effective than the Functional Test Set I in verification of the ALU because the results of arithmetic and logic operations are often placed in a register without being sensitized to the outputs. It is necessary to increase the effectiveness of the test by adding vectors to sensitize the results of these operations. As an example, the following sequence occurs in Vectors 578 through 587.

MNEMONIC	DATA CONTAINED IN ACCUMULATOR B	OPERATION
	FF (11111111)	
ROR B	7F (01111111)	Rotate right B
LSR B	3F (00111111)	Logical shift right B
ASR B	1F (00011111)	Arithmetic shift right B
ROL B	3F (00111111)	Rotate left B
CLR B	00 (00000000)	Clear B

In this sequence four successive shift operations are performed on Accumulator B. At no point during this sequence is the content of B nor the content of the Condition Code Register verified. After the shifts, a CLEAR B is executed which destroys the contents of both registers, thereby destroying all record of these operations. This is the only occurrence of these four shifts on Register B, therefore, these four instructions are never verified. By the addition of an instruction such as STA B (store Register B), before the CLR B, the results of the shifting circuitry is checked. It is not normally recommended that a vector be inserted in an existing test set as this may upset an attempt by the original programmer to check for a known pattern sensitivity. However, in this case and in other similar cases, the STA B was inserted since a large number of vectors would otherwise have to be added to reproduce the same conditions present after the ROL B. Also the chances of pattern sensitivity at this point in the sequence are slim. In addition, this series of instructions exemplifies a risky test technique. By repeatedly pushing data through the same circuitry (in this case the right shift circuitry) without monitoring the intermediate data contained in the B Register, a fault may mask itself. That is, a fault may cause a failure to occur and subsequent instructions may cause a complementing failure which would cause the final result to be the same as that produced by a good circuit. Since many of the faults are caught elsewhere with the recommended tests these vectors were not changed. Also a large increase in the number of vectors would result for a marginal increase in testing confidence.

Tables 3-1 through 3-7 delineate the faults detected in the ALU by Vendor A's vectors plus those vectors that were added. The nomenclature used in the tables follows:

X - Detected with Vendor A's vectors.
I - Conditions set-up with Vendor A's vectors but required insertion of sensitizing vectors to detect internal states.
A_n - Detected with added vectors not including sensitizing vectors as in (I) above. A_n - Refers to the Add-1 through Add-12 series of appended vectors.

Input Conditions			Bit Position								Carry		
Carry	DI_1	DI_2	7	6	5	4	3	2	1	0	Function	Value	
0	0	0	A_{12}	X	X	A_{12}	A_{11}	X	X	X	Carry In Add	0 1	A-12 A-9
0	0	1	A_{13}	X	X	X	X	X	X	X	Carry In Subtract	0 1	X A-11
0	1	0	X	A_{12}	X	X	X	A_{14}	X	A_{11}	Carry Out	0 1	X X
0	1	1	X	X	X	A_{11}	X	A_{12}	A_{11}	X			
1	0	0	X	A_{11}	X	X	X	A_{11}	X	A_{14}			
1	0	1	X	X	X	X	X	X	A_{13}	X			
1	1	0	I	X	I	X	I	X	X	A_{13}			
1	1	1	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9			

Table 3-1. Conditions Applied to ALU in ADD Mode

Input Condition		Bit Position							
DI_1	DI_2	7	6	5	4	3	2	1	0
0	0	A_2	A_2	A_2	A_2	A_2	A_2	X	A_2
0	1	I	I	I	I	X	I	I	I
1	0	A_3	A_3	X	X	A_3	A_3	A_2	X

Table 3-2. Conditions Applied to ALU in OR Mode

Input Condition		Bit Position							
DI_1	DI_2	7	6	5	4	3	2	1	0
0	1	X	X	X	X	A_7	X	X	X
1	0	A_7	A_7	A_7	A_7	X	A_7	A_7	A_7
1	1	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}

Table 3-3. Conditions Applied to ALU in AND Mode

Input Condition		Bit Position							
DI_1	DI_2	7	6	5	4	3	2	1	0
0	0	A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6
0	1	A_8	A_8	A_8	A_8	I	A_8	I	A_8
1	0	I	I	I	I	A_8	I	A_8	I
1	1	I	I	I	I	I	I	I	I

Table 3-4. Conditions Applied to ALU in EXOR Mode

Input Condition	Bit Position							
	7	6	5	4	3	2	1	0
0	X	X	X	X	X	X	X	X
1	X	A_5	X	A_5	X	A_5	X	A_5

Table 3-5. Conditions Applied to the One's Complementor

Input Condition	Bit Position								Force a "0" With Carry = "1"	Carry-In		Carry Out		ARITH SHIFT	
	7	6	5	4	3	2	1	0		0	1	0	1	0	1
0	X	X	X	X	X	X	X	X	X	X	X	X	I	X	A_1
1	X	X	X	X	X	X	X	X							

Table 3-6. Conditions Applied to Shift Right

Input Condition	Bit Position								Force a "0" With Carry = "1"	Carry-In		Carry Out	
	7	6	5	4	3	2	1	0		0	1	0	1
0	I	X	I	X	I	X	I	X	X	X	I	I	X
1	X	I	X	I	X	I	X	I					

Table 3-7. Conditions Applied to Left Shift

3.3.5 Register Array

The Register Array in the Mc6800 consists of the following registers:

1) Two 8-bit accumulators - These are the most powerful registers and can be used in any arithmetic or logic instruction and for temporary data storage.
2) One 16-bit Index Register used for indexed address modification and temporary data storage.
3) One 16-bit Stack Pointer - This is a special purpose register used to identify the area of RAM where the other registers are stored during any interrupt or subroutine.
4) One 16-bit Program Counter - This is a specialized register which points to the next instruction to be executed at each point in a program.
5) A 6-bit Condition Code Register which stores the Half-Carry, Interrupt Mask, Negative, Zero, Overflow and Full-Carry conditions from previous operations and is used in Conditional Branch operations.
6) Two 8-bit temporary registers associated with the data and address buses.
7) And also an Incrementor/Decrementor which operates on the three 16-bit registers and the address buffer latches.

A register array is a random access memory and, as such, lends itself to tests using many of the standard tests developed for RAMs. For example, it is possible to apply a modified Walking 1/0 and Galloping 1/0 patterns to most of the register array. The Vendor A Functional Test Sets did not include such patterns. The designers of the Mc6800, who also developed the test vectors, were contacted and questioned as to why these tests were not included. They explained that the register array was purposely built out of static latches over a large area of the chip so that the pattern and temperature sensitivities associated with RAMs would be eliminated.

For those testers with algorithmic hardware pattern generators, walking 1/0 and galloping 1/0 patterns could be applied to the register arrays with relatively few test instructions. Since a truth table representation of these patternsis prohibitively large, and since Vendor A asserts that little is gained by these patterns, these patterns are not included nor required as part of the recommended test set. If it is later determined that this type of pattern detects a significant number of faulty chips which are missed by the rest of the test, then they will be added at that time.

Vendor A specifies a minimum clock frequency for the processor. This is because the buses were designed to operate dynamically, i.e., they depend on capacitance for data transfer. Based on this assumption, the evaluation approach was to check for:

1) register independence,
2) bit independence and
3) flip-flop integrity (i.e. insure transitions of 0 to 0, 0 to 1, 1 to 1 and 1 to 0 for each bit in each register).

The registers are uniquely specified several times. This occurs whenever an interrupt occurs and the data in each eight-bit section of the register array, (i.e. each single length register and each half of the double length registers) is unique. An example of this begins at Vector 192 in the long test (Functional Test Set I). The registers were previously loaded with the following:

Register	Data
PC (Low)	C9
PC (High)	OD
INDX (Low)	OO
INDX (High)	96
A	81
B	11
Flag	FO

When the op code, SWI (Software Interrupt) is executed, the contents of the registers are sequentially output to the data bus. If the register select circuitry is faulty, the correct data will not be output in the correct order.

In reviewing Functional Test Set I, it was found that many faults could be detected by:

1) counting up by one from all ones (all bits change from 1 to 0), and
2) counting down by one from all zeroes (all bits change from 0 to 1).

Note, however, that these transitions do not check for bit independence. This method was used on most of the registers in Functional Test Set II. Unfortunately, very little else was checked on several of the registers particularly with respect to bit independence. Following is a description of the types of tests performed or not performed on each register using Functional Test Set II. The particular tests performed on each register are detailed in Tables 3-8 to 3-13.

	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X
0 0	I	I	I	I	I	I	I	I
0 1	I	I	I	I	I	I	I	I
1 0	I	I	I	I	I	I	I	I
1 1	A 16	A 12	A 11	A 11	A 10	A 12	A 11	A 11

BIT AT ZERO

BIT AT ONE

0	A 11	X	X	X	X	A 11	X	
1	A 11	X	X	X	X	A 11		A 15
2	I	X	X	X	X		X	I
3	X	X	X	A 11		X	X	X
4	X	X	X		X	X	X	X
5	A 11	A 11		A 11	X	A 11	X	A 15
6	I		I	A 11	X	A 12	X	I
7		X	X	X	X	X	X	X

Table 3-8. Accumulator A

	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X
0 0	I	I	I	I	I	I	I	I
0 1	I	I	I	I	I	I	I	I
1 0	X	I	I	I	I	I	I	I
1 1	A 12	I	I	I	I	I	I	I

BIT AT ZERO

BIT AT ONE

0	I	I	I	I	I	I	I	
1	I	I	I	A 2	A 2	A 2		A 2
2	I	I	I	A 3	I		I	I
3	I	I	I	A 3		A 12	A 3	A 3
4	I	I	I		I	A 12	I	I
5	I	I		A 6	A 6	A 12	A 3	A 3
6	I		I	A 3	I	A 12	I	I
7		A 13	A 3	A 3	A 6	A 13	A 3	A 3

Table 3-9. Accumulator B

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 1	A 3	A 6	A 3	A 6	A 3	A 6	A 3	A 6	A 3	A 6	A 3	A 6	A 3	A 6	A 3	A 6

BIT AT ZERO

BIT AT ONE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	A 6	A 8	A 10	A 6	A 8	A 11	A 6	A 11	A 8	A 8	A 10	A 10	A 8	A 8	A 12	
1	A 6	A 3	A 10	A 3	A 8	A 3	A 6	A 3	A 8	A 3	A 10	A 3	A 8	A 3		A 3
2	A 6	A 10	A 10	A 6	A 11	A 11	A 6	A 11	A 10	A 10	A 10	A 10	A 12		A 12	A 13
3	A 6	A 3	A 10	A 3	A 11	A 3	A 6	A 3	A 10	A 3	A 10	A 3		A 3	A 9	A 3
4	A 6	A 8	A 11	A 6	A 6A	A 6A	A 6	A 6A	A 8	A 8	A 13		A 6A	A 6A	A 6A	A 6A
5	A 6	A 3	A 11	A 3	A 6A	A 3	A 6	A 3	A 8	A 3		A 3	A 6A	A 3	A 6A	A 3
6	A 6	A 11	A 11	A 6	A 6A	A 6A	A 6	A 6A	A 12		A 13	A 12	A 6A	A 6A	A 6A	A 6A
7	A 6	A 3	A 11	A 3	A 6A	A 3	A 6	A 3		A 3	A 13	A 3	A 6A	A 3	A 6A	A 3
8	A 6	A 8	A 10	A 6	A 8	A 13	A 6		A 8	A 8	A 9	A 9	A 8	A 8	A 9	A 9
9	A 10	A 3	A 10	A 3	A 8	A 3		A 3	A 8	A 3	A 9	A 3	A 8	A 3	A 9	A 3
10	A 6	A 10	A 10	A 6	A 12		A 6	A 12	A 9	A 9	A 9	A 9	A 9	A 9	A 9	A 9
11	A 6	A 3	A 10	A 3		A 3	A 6	A 3	A 4	A 3	A 9	A 3	A 9	A 3	A 9	A 3
12	A 6	A 8	A 12		A 6A	A 6A	A 6	A 6A	A 8	A 8	A 9	A 9	A 6A	A 6A	A 6A	A 6A
13	A 6	A 3		A 3	A 6A	A 3	A 6	A 3	A 8	A 3	A 9	A 3	A 6A	A 3	A 6A	A 3
14	A 6		A 12	A 6	A 6A	A 6A	A 6	A 6A	A 9	A 9	A 9	A 9	A 6A	A 6A	A 6A	A 6A
15		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3-10. Index

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 0	A 14	A 12	A 14	A 12	A 13	A 15	A 13	A 15	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 1	A 6	A 14	X 6	X 14	X 6	X 13	X 6	X 13	X	X	X	X	X	X	X	A 12

BIT AT ZERO

BIT AT ONE

0	A 13	A 6	A 13	A 6	A 12	A 6	A 12	A 6	A 12	X	X	X	X	X	X	
1	A 13	A 6	A 13	A 6	A 12	A 6	A 12	A 6	A 12	X	X	X	X	X		X
2	A 13	A 6	A 13	A 6	A 12	A 6	A 12	A 6	A 12	X	X	X	X		X	X
3	A 13	A 6	A 13	A 6	A 12	A 6	A 12	A 6	A 12	X	X	X		X	X	X
4	A 13	A 6	A 13	A 6	A 12	A 6	A 12	A 6	A 12	X	X		X	X	X	X
5	A 13	A 6	A 13	A 6	A 12	A 6	A 12	A 6	A 12	X		X	X	X	X	X
6	A 13	A 6	A 13	A 6	A 12	A 6	A 12	A 6	A 12		X	X	X	X	X	X
7	A 14	A 6	A 14	A 6	A 12	A 6	A 12	A 6		X	X	X	X	X	X	X
8	A 13	A 12	A 13	A 12	A 12	A 12	A 12		A 12	X	X	X	X	X	X	X
9	A 14	A 6	A 14	A 6	A 16	A 6		A 6	A 16	X	X	X	X	X	X	X
10	A 13	A 12	A 13	A 12	A 12		A 12	A 12	A 12	X	X	X	X	X	X	X
11	A 14	A 6	A 14	A 6		A 6	A 15	A 6	A 15	X	X	X	X	X	X	X
12	A 13	A 15	A 13		A 13	A 14	A 13	A 14	A 13	X	X	X	X	X	X	X
13	A 14	A 6		A 6	A 12	A 6	A 12	A 6	A 12	X	X	X	X	X	X	X
14	A 13		A 13	A 16	A 13	A 14	A 13	A 14	A 13	X	X	X	X	X	X	X
15		A 6	A 16	A 6	A 12	A 6	A 12	A 6	A 12	X	X	X	X	X	X	X

Table 3-11. Stack Pointer

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 0	X	X	X	X	X	X	X	I	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

BIT AT ZERO

BIT AT ONE																	
	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
	2	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X
	3	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X
	4	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X
	5	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X
	6	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X
	7	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X
	8	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X
	9	X	X	X	X	X	X		x	X	X	X	X	X	X	X	X
	10	A 3	A 7	A 3	A 9	A 3		A 3	A 10	X	X	X	X	X	X	X	X
	11	A 7	A 2	A 7	A 2		A 2	A 5	A 2	X	X	X	X	X	X	X	X
	12	A 3	A 7	A 3		A 3	A 8	A 3	A 10	X	X	X	X	X	X	X	X
	13	A 9	A 2		A 2	A 9	A 2	A 9	A 2	X	X	X	X	X	X	X	X
	14	A 3		A 3	A 9	A 3	A 8	A 3	A 5	X	X	X	X	X	X	X	X
	15		A 2	A 10	A 2	A 7	A 2	A 5	A 2	X	X	X	X	X	X	X	X

Table 3-12. Program Counter

		H	I	N	Z	V	C
DATA	= 0	I	X	X	X	X	X
DATA	= 1	X	X	X	X	X	X
0	0	I	I	X	X	X	I
0	1	A_{15}	X	X	X	X	X
1	0	A_{16}	X	X	X	X	X
1	1	X	X	X	X	X	X

		BIT AT ZERO					
BIT AT ONE	C	A_{16}	I	X	X	X	
	V	I	I	X	X		A_{15}
	Z	I	I	X		X	I
	N	I	I		X	X	X
	I	X		X	X	X	X
	H		X	X	X	X	X

Table 3-13. Condition Code Register

Most register bits were verified as retaining a logic 1 and logic 0, otherwise the over-all conclusions were as follows:

The A and B Registers were exercised extensively. However, the contents of these registers were verified relatively infrequently with respect to the frequency of operations on each register. This makes verification of some of the tests virtually impossible because repeated operations may mask faults.

The tests which are indicated as verified in Tables 3-8 and 3-9 are those faults which are not likely to be masked by repeated operations on the registers.

The only two values loaded into the INDEX Register are 0000_{16} and $FFFF_{16}$. No attempt was made to check bit independence within the INDEX Register.

The patterns which are applied to the upper eight bits of the STACK POINTER and the upper six bits of the PROGRAM COUNTER are the same as those applied to the INDEX Register. Bit independence is not checked between these bits. In the lower order bits, most faults were detected. The faults are shown in Tables 3-11 and 3-12.

The test on the CONDITION CODE Register was not complete. The results are detailed in Table 3-13.

The INSTRUCTION Register was not analyzed as part of the register array but is included as an integral part of the Instruction Decode Circuitry; i.e. The Instruction Register is verified by insuring that all instructions are executed correctly.

Vectors were inserted or added to catch faults not detected in Functional Test Set II. Tables 3-8 through 3-13 indicate whether the faults were detected by inserted vectors or by the vectors added to the end of the test set. For vectors added to the end, the number (ADD-1 through ADD-6) specifies the group or series of vectors in which the faults are detected in the final test set. The nomenclature used in the tables is as follows:

X - Detected with Vendor A's vectors.

I - Conditions set-up with Vendor A's vectors but required insertion of sensitizing vectors to detect internal states.

A_n - Detected with added vectors not including sensitizing vectors as in (I) above.

3.3.6 Input and Output Signal Specifications

Since the majority of a uP's circuitry is buried and has limited access from the device pin, it is necessary to test the uP at clock rates to obtain a good test of the uP's dynamic characteristics. In addition, the large number of test vectors and the requirement for precise input conditions necessitates the use of a high-speed, versatile tester. It is practical, and in many cases necessary, to combine input threshold, output level, dynamic and switching tests with the functional test.

Five sets of input conditions were chose to dynamically test the uP. They are specified in Figure 3-10 and graphically shown in Figures 3-11 through 3-24. Each column (Condition A through E) specifies the input conditions to be applied during a pass through the functional test set.

The input voltage levels were chosen with V_{IH} equal to the minimum (V_{cc} - 0.3V for the clocks and 2.0V for all others) and the V_{IL} equal to the maximum (0.3V for the clocks and 0.8V for all others). The waveforms and the setup and hold times for the inputs are as follows:

a. Period - The microprocessor is a cyclical device and as such some of the inputs and all outputs are referenced to these repeating waveforms at some point(s) during each cycle. Therefore, the strobe points for these signals are also repeating at the same rate as the clocks. The rate of repetition is referred to as the period. 1 us and 10 us were chosen as the two test periods because they represent the highest (1 MHZ) and lowest (100 KHZ) operating frequencies for the 6800. The first ensures that the internal propagation delays permit maximum frequency operation and the second ensures that leakage in the dynamic portions of the device is low enough such that the uP will run at minimum frequency.

b. Clocks - The clock input timing specifications are pulse width (time and logic one) for each clock input (t_{WH1}, t_{WH2}), and the delay from the trailing edge of Ø1 to the leading edge of Ø2 ($t_{SØ1}$, $_{Ø2}$) or vice versa ($t_{SØ2}$, $_{Ø1}$). At the high frequency (period = 1 us) with the minimum clock pulse widths, a moderate amount of rise and fall time is permitted. This means that $t_{S(Ø1, Ø2)}$ and $t_{S(Ø2, Ø1)}$ are both approximately equal to zero. Two sets of input conditions (Conditions A and B) both include this set of conditions; one is performed with V_{cc} = 4.75V and the other with V_{cc} = 5.25V.

At lower frequencies there is a great deal of flexibility in the clock widths and phase. Three conditions were chosen at the lowest frequency (100 KHZ). Two of these conditions (Conditions C and D) specify minimum high level pulse widths on both clocks. This permits maximum separation from the falling edge of one clock to the rising edge of the other clock. This is significant for testing the dynamic portions of the 6800. The address and data buses, and the timing circuitry operate by charge transfer and storage. A test of this type of circuitry should check that sufficient charge is transferred in the minimum transfer time and that sufficient charge is stored with maximum storage time for proper operation. From telecons with Vendor A personnel and from the waveform/timing diagrams in the Mc6800 Microcomputer System Design Data book, the operation of the dynamic buses were determined to be as follows:

Read Cycle

- The Ø2 clock "1" level enables data into the device and the data bus is charged during this time. Note that the leading edge of Ø2 always precharges the bus to a "1" level. If data is other than a "1", the bus is pulled low by draining off the charge.

- The Ø2 clock trailing edge disconnects the bus from the device pins.

- The leading edge of Ø1 transfers the data bus information in internal registers.

Write Cycle

- A low level on DBE combined with a high level on Ø1 enables data from internal registers to the data bus buffers through a "coupling device". The rising edge of DBE will clock the data to the external pins. With input Conditions C and D the minimum Ø1 high level (combined with DBE low) tests the data bus charge transfer during write cycles and the minimum Ø2 high level tests the data bus charge transfer during read cycles. Condition C applies the clocks with a maximum Ø2 to Ø1 delay. This condition allows minimum time between the transfer of data to the internal registers and the precharging of the data bus in preparation for the next cycle. This test condition was the only one to fail one of the sample devices; however, the failure mechanism may not necessarily be the one described above. Input Condition D applies the clocks with a maximum Ø1 to Ø2 delay. This allows the maximum time for the data bus to discharge and therefore tests for the failures due to data bus leakage. Input Condition E specifies the maximum duty cycle on both clocks at minimum frequency.

c. DBE - The Data Bus Enable (DBE) pin, in part, controls the time for which output data is stable during write cycles. The minimum time DBE must be low in order to transfer the data is 150 us. However, at least 150 us of the DBE low time must be coincident with Ø1 high and must end at least 280 us after the rising edge of Ø1. By applying DBE low from 130 us until 280 us after the rising edge of Ø1, the worst case charge time for write data is applied. This tests the charge transfer circuitry It also provides the maximum valid time for write data. These conditions are applied in Input Conditions A, B and C.

The maximum DBE low time is when DBE equals Ø2. This condition tests that leakage from the Data Bus source is small enough during charge transfer to allow normal operation with a long storage time. This occurs in Input Conditions D and E.

d. Data Bus (Input Mode) - On read cycles, the data bus must be held stable from 100 ns prior to, until 10 us after, the trailing edge of Ø2. The data inputs are specified as driven and stable at the values specified in Appendix A during every read cycle and also specified as driven to the opposite value before and after the above read period during every read cycle.

e. $\overline{\text{Halt}}$ - The $\overline{\text{Halt}}$ input was verified for two input conditions for Vectors 826 through 835, the minimum low level time is specified. This checks that the uP will enter the halt mode with minimum low level setup and hold times. In Vectors 1292 through 1295 and 1300 through 1308 the minimum high level time is specified. This checks that the uP will leave the wait state with minimum high level setup and hold times.

f. TSC - The tri-state control (TSC) line puts the memory interface pins (data, address and R/W) in the high impedance state when the line is forced high. In normal operation TSC is brought high while Ø1 is high and Ø2 is low (Figure 3-21). This operation is verified in Conditions C, D and E. However, since there is a 0 us to 700 us delay on the outputs, there is not sufficient time at the higher clock rate (Conditions A and B) for the outputs to be guaranteed of entering the high impedance state. In order to retain the same set of vectors for each of the input conditions. TSC is applied for all input conditions. However the outputs should not be monitored for high impedance for Input Conditions A and B (Figure 3-9).

g. Output Conditions - The output timing and voltage conditions shown in Figure 3-10 are as specified by Vendor A as the worst case conditions for the inputs specified.

Figures 3-11 through 3-22 show the timing relationships between the various inputs and outputs. Figures 3-9 and 3-10 reference 3-11 through 3-22 for each setup or measurement condition and also specify the values for the conditions.

		INPUT CONDITIONS						
SYMBOL	TERMINAL(S)	A	B	C	D	E	UNITS	FIGURE
V_{CC}	V_{CC}	5.25	4.75	4.75	4.75	4.75	Volts	-
V_{IH1}	Logic Inputs	2.0	2.0	2.0	2.0	2.0	Volts	-
V_{IL1}	Logic Inputs	0.8	0.8	0.8	0.8	0.8	Volts	-
V_{IH2}	Ø1, Ø2	4.95	4.35	4.35	4.35	4.35	Volts	-
V_{IL2}	Ø1, Ø2	0.3	0.3	0.3	0.3	0.3	Volts	-
t_{Period}	Ø1	1	1	10	10	10	μs	3-11
t_{WH1}	Ø1	430	430	430	430	4500	ns	
t_{WH2}	Ø2	450	450	450	450	4500	ns	
t_S(Ø1,Ø2)	Ø2	0	0	0	--	0	ns	
t_S(Ø2.Ø1)	Ø2	--	--	--	0	--	ns	
t_{WL1}	DBE	150	150	150	DBE	DBE	ns	
t_{SLH1}	DBE	280	280	280	EQUALS	EQUALS	ns	
t_{SHL1}	DBE	300	300	300	Ø2	Ø2	ns	3-11

Figure 3-9. Input Conditions

(Continued on next page)

		INPUT CONDITIONS						
SYMBOL	TERMINAL(S)	A	B	C	D	E	UNITS	FIGURE
t_{SLH4}	Data (In)	225	225	225	225	225	ns	3-13
t_{SHL4}	Data (In)	225	225	225	225	225	ns	
t_{HHL4}	Data (In)	10	10	10	10	10	ns	
t_{HLH4}	Data (In)	10	10	10	10	10	ns	3-13
t_{WH4}	$\overline{\text{Halt}}$	200	-	200	-	-	ns	
t_{WL2}	$\overline{\text{Halt}}$	200	-	200	-	-	ns	3-19
t_{SHL_3}	$\overline{\text{Halt}}$	-	200	-	200	200	ns	
t_{SLH3}	$\overline{\text{Halt}}$	-	200	-	200	200	ns	3-19
t_{SLH2}	$\overline{\text{Halt}}$, $\overline{\text{Reset}}$ $\overline{\text{NMI}}$. $\overline{\text{IRQ}}$	200	200	200	200	200	ns	3-16 thru 3-19
t_{SHL2}	$\overline{\text{Halt}}$. $\overline{\text{Reset}}$ $\overline{\text{NMI}}$ and $\overline{\text{IRQ}}$	200	200	200	200	200	ns	3-17 thru 3-19
t_{SLH3}	TSC	40	40	40	40	400	ns	
t_{WH3}	TSC	*560	*560	9560	9560	2000	ns	

*Ignore high impedance outputs resulting from TSC on test conditions (A) and (B).

Figure 3-9. Input Conditions

(Concluded)

SYMBOL	MEASUREMENT TERMINAL	FIGURE	TEST LIMITS						UNITS
			Subgroup 7 T_A = 25°C		Subgroup 8 T_A = -40°C		Subgroup 8 T_A = +25°C		
			Min	Max	Min	Max	Min	Max	
V_{OH}	All Outputs	-	2.4	---	2.4	---	2.4	---	V
V_{OL}	All Outputs	-	---	0.4	---	0.4	---	0.4	V
t_{PLH1}	R/W, A0-A15, VMA	3-13, 3-14					---	300	ns
t_{PHL1}	R/W, A0-A15						---	300	ns
t_{PLH2}	R/W, A0-A15						50	---	ns
t_{PHL2}	R/W, A0-A15, VMA	3-13, 3-14					50	---	ns
t_{PZH1}	D0-D7	3-15					---	225	ns
t_{PZL1}	D0-D7						---	225	ns
t_{PHZ1}	D0-D7						10	---	ns
t_{PLZ1}	D-D7	3-15					10	---	ns
t_{PHZ2}	A0-A15						---	700	ns
t_{PLZ2}	R/W						---	700	ns
t_{PZH2}							---	700	ns
t_{PZL2}							---	700	ns
t_{PLH3}	VMA, BA						---	50	ns
t_{PHL3}	VMA, BA						---	50	ns
t_{PLH4}	BA	3-19					---	300	ns
t_{PHL4}	BA						---	300	ns
t_{PZH3}	R/W, 0-A15	3-20					---	300	ns
t_{PZL3}	A0-A15						---	300	ns
t_{PLZ3}	R/W, A0-A15						50	---	ns
t_{PHZ3}	R/W, A0-A15	3-20					50	---	ns

Figure 3-10. Functional Test Output Conditions
(For Input Conditions A-E, Figure 3-9)

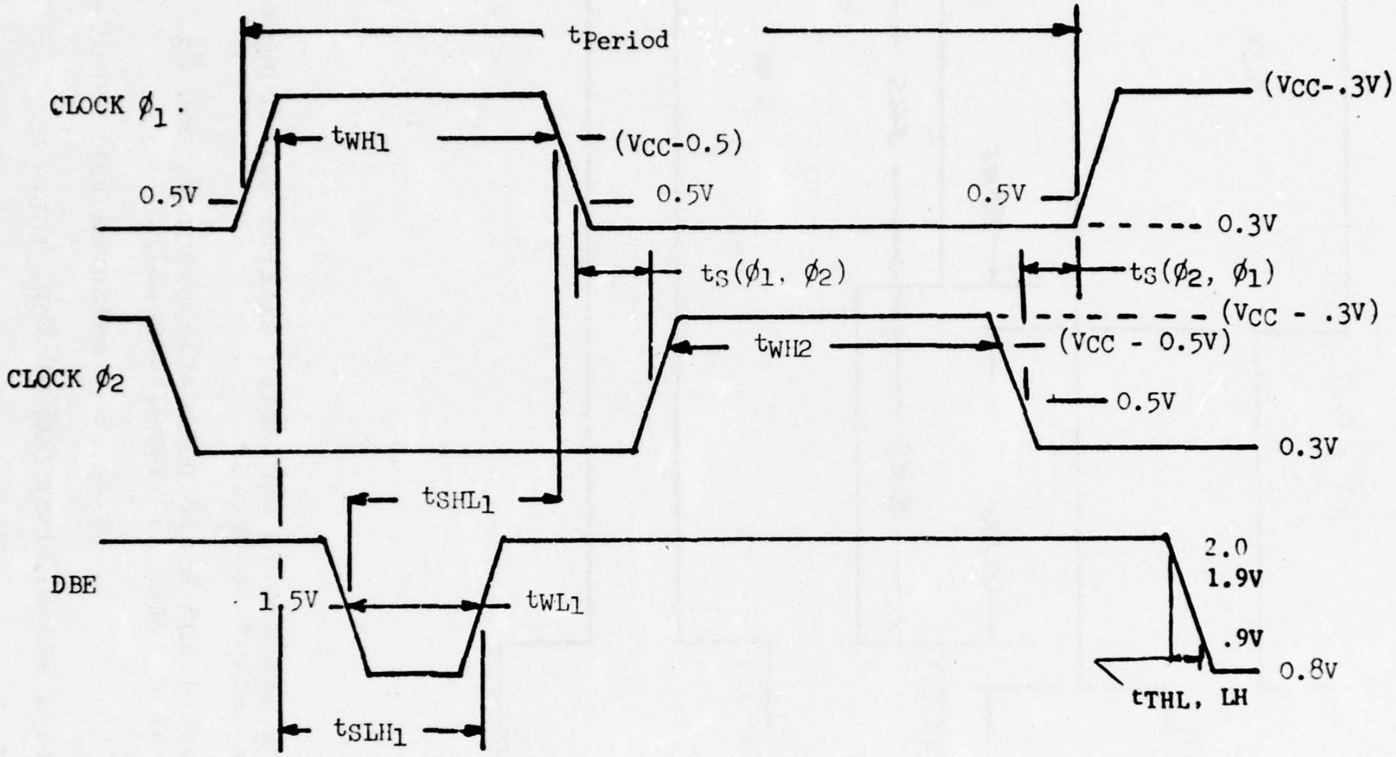
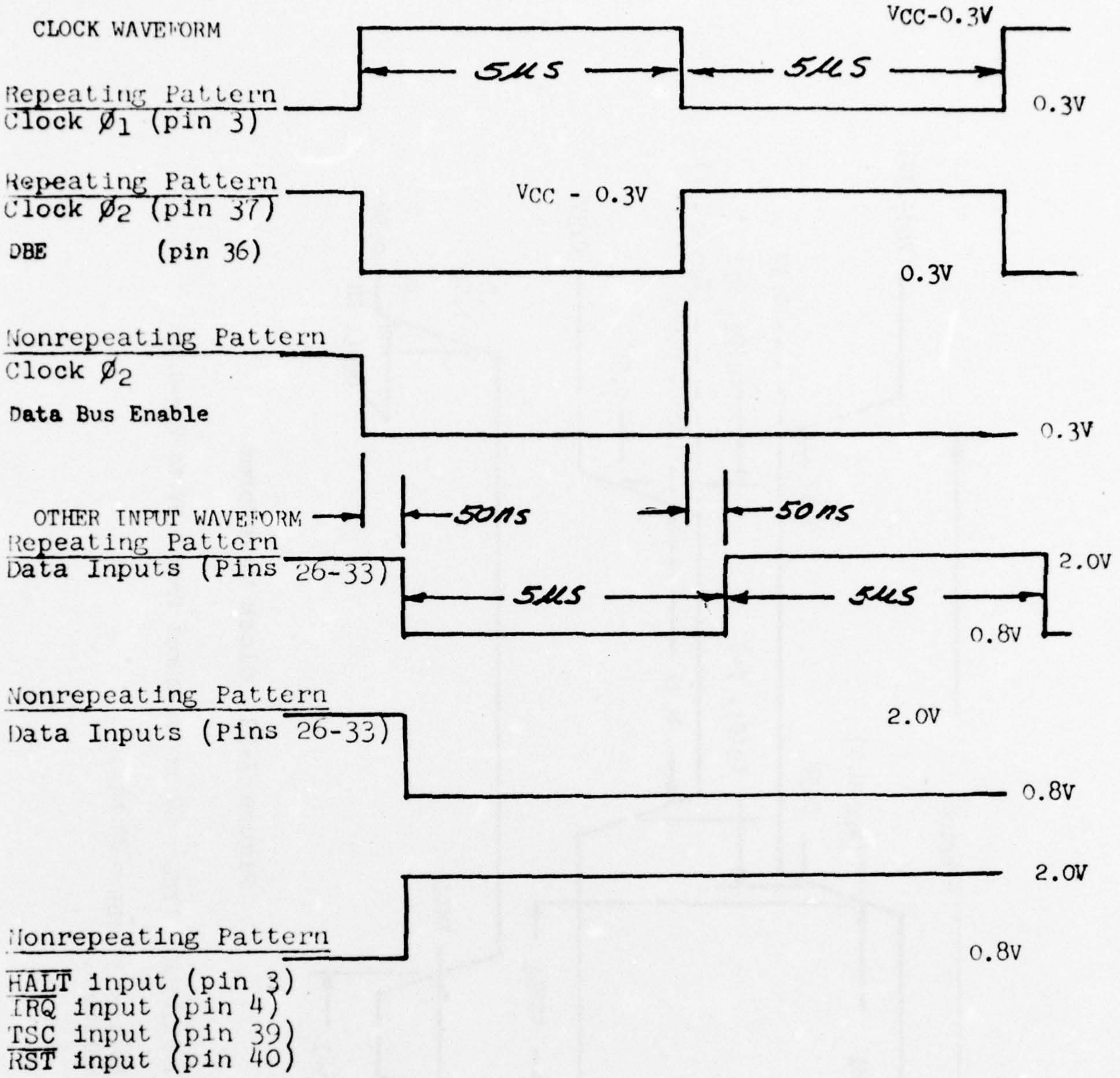


Figure 3-11. Clock Waveforms

NOTES:

1. Clock, $t_{TLH} = t_{THL}$ = 50 ns; measured from 0.5V to V_{CC}-0.5V.

2. DBE, $t_{TLH} = t_{THL}$ = 25 ns; amplitude.



NOTES:

1. A repeating pattern is one which alternately changes state from vector to vector.

2. Clock phases 1 and 2 are nonoverlapping; $\emptyset_1$ and $\emptyset_2$ cannot be at a logic "1" simultaneously.

3. Tr = tf = 50 ns measured between 10-90%.

4. Timing delays measured at 50% of V_{OH}, V_{OL}.

5. $V_{OH} = 2.4V$, $V_{OL} = 0.4V$.

Figure 3-12. Tri-state Preconditioning Pattern Waveforms

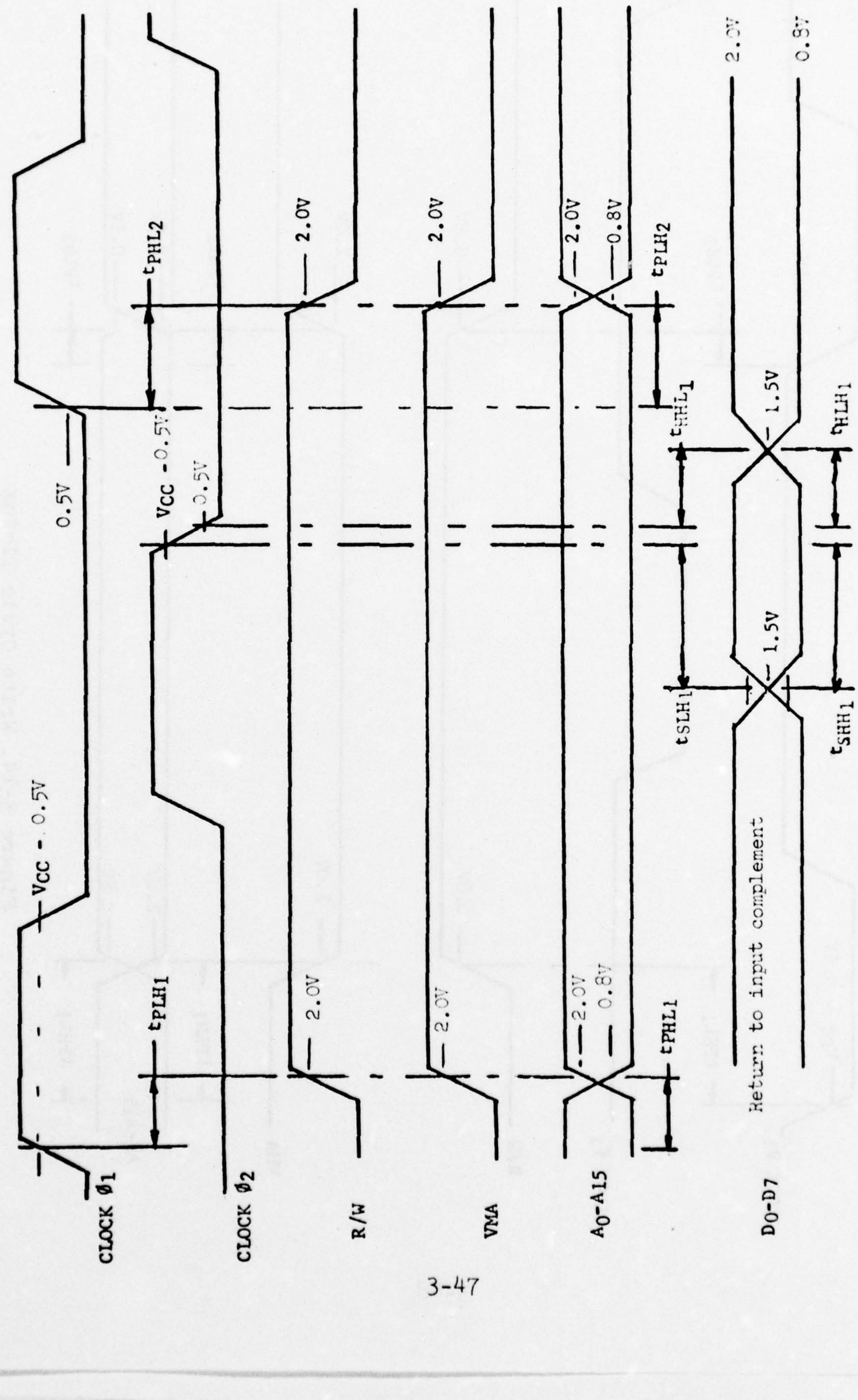


Figure 3-13. Read Cycle Timing

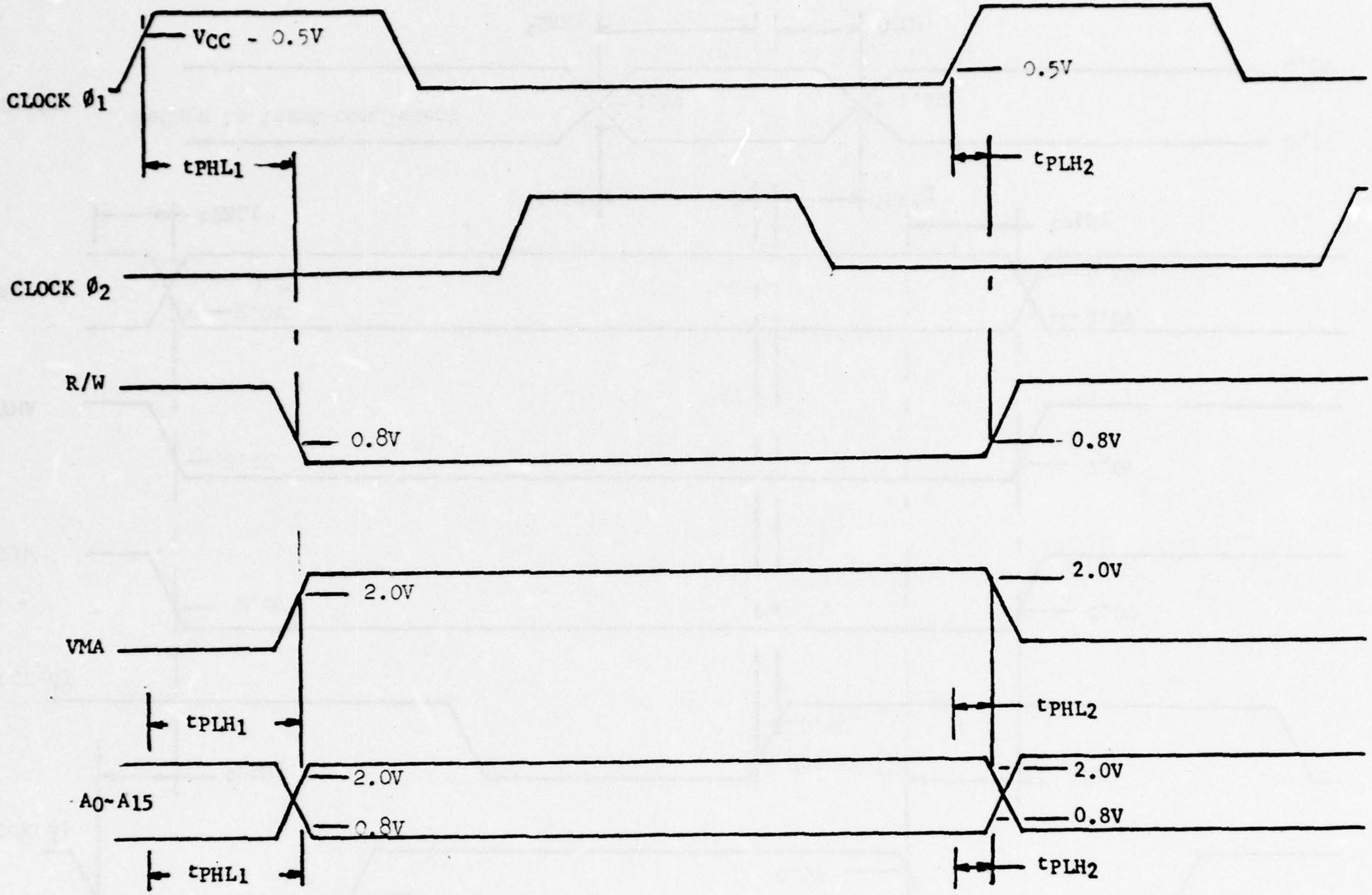


Figure 3-14. Write Cycle Timing

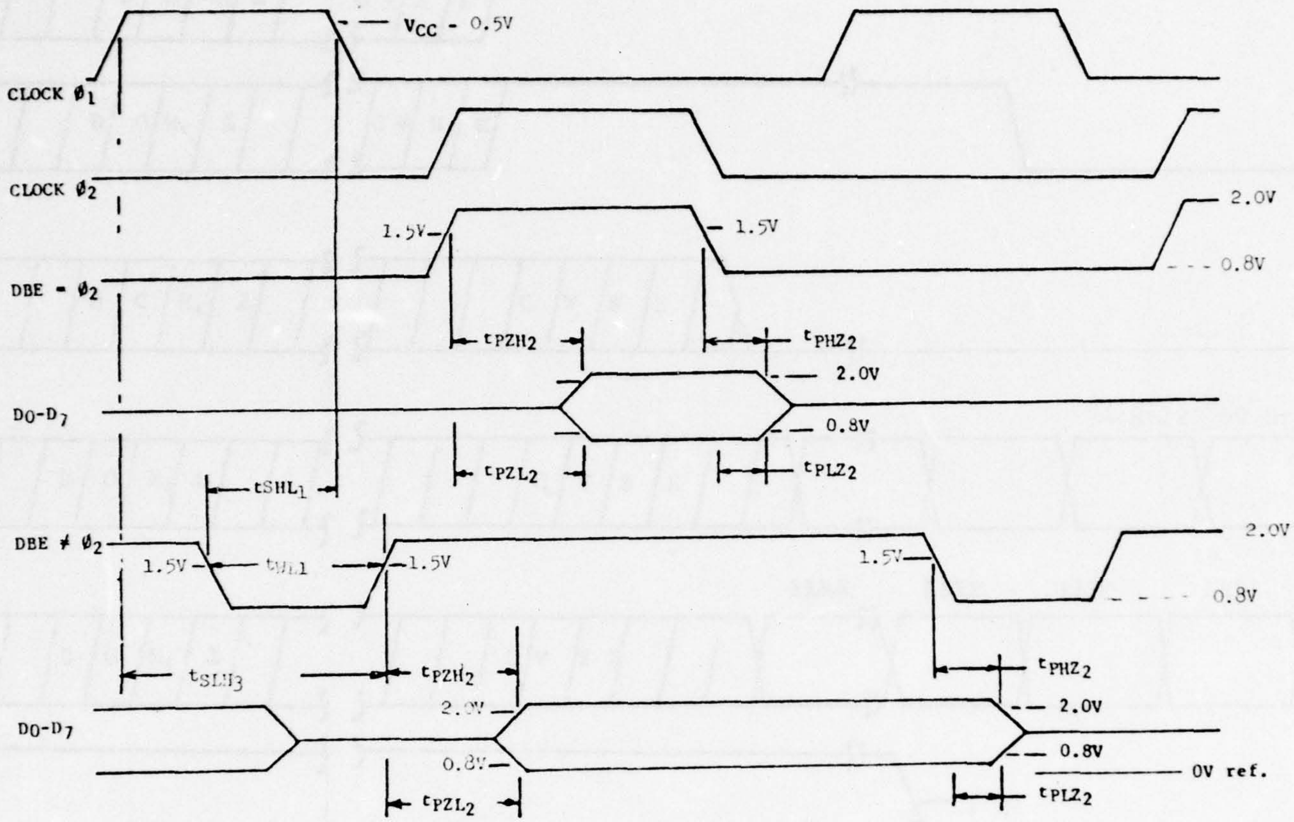


Figure 3-15. Write Cycle Timing

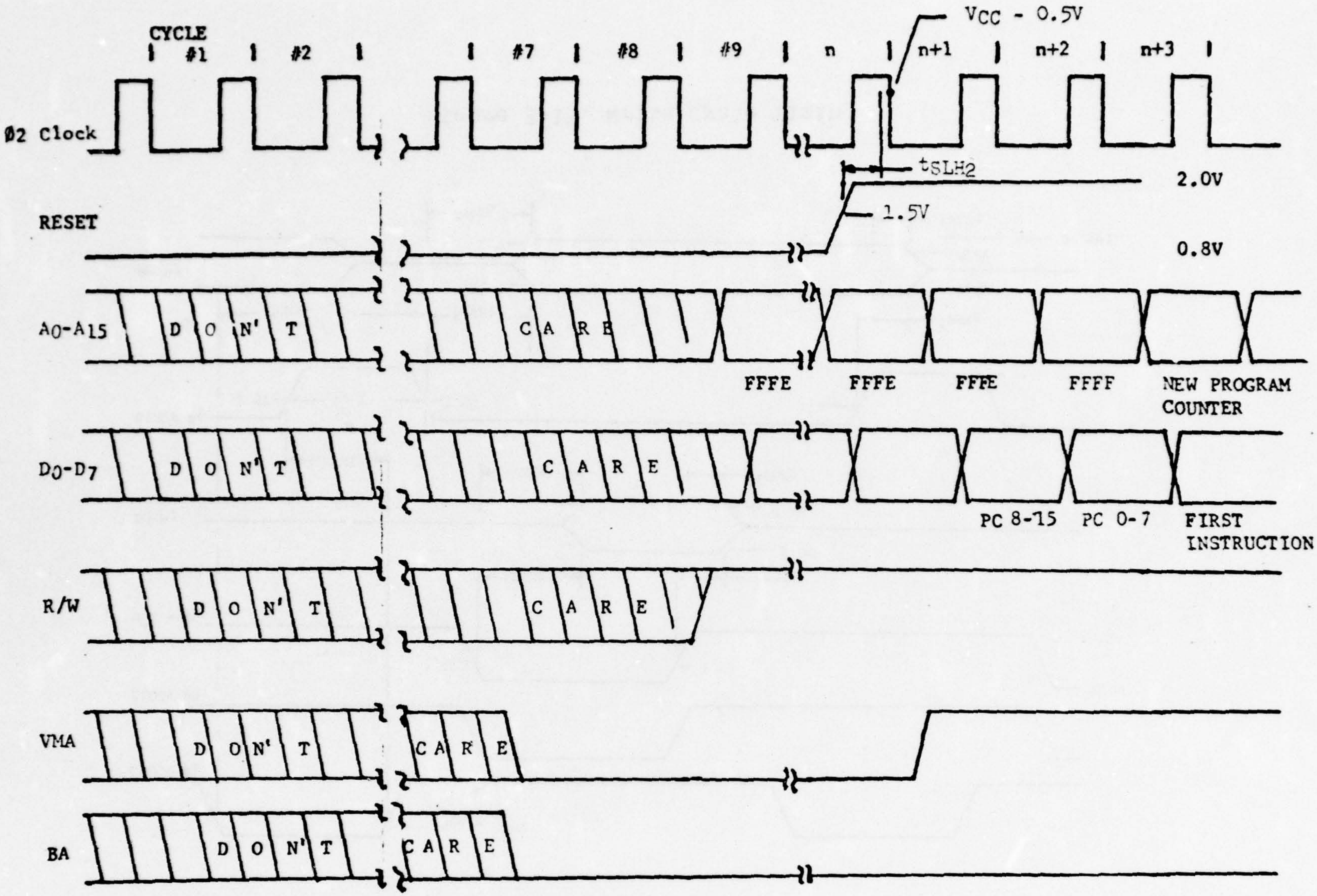


Figure 3-16. Power Up-Reset Timing

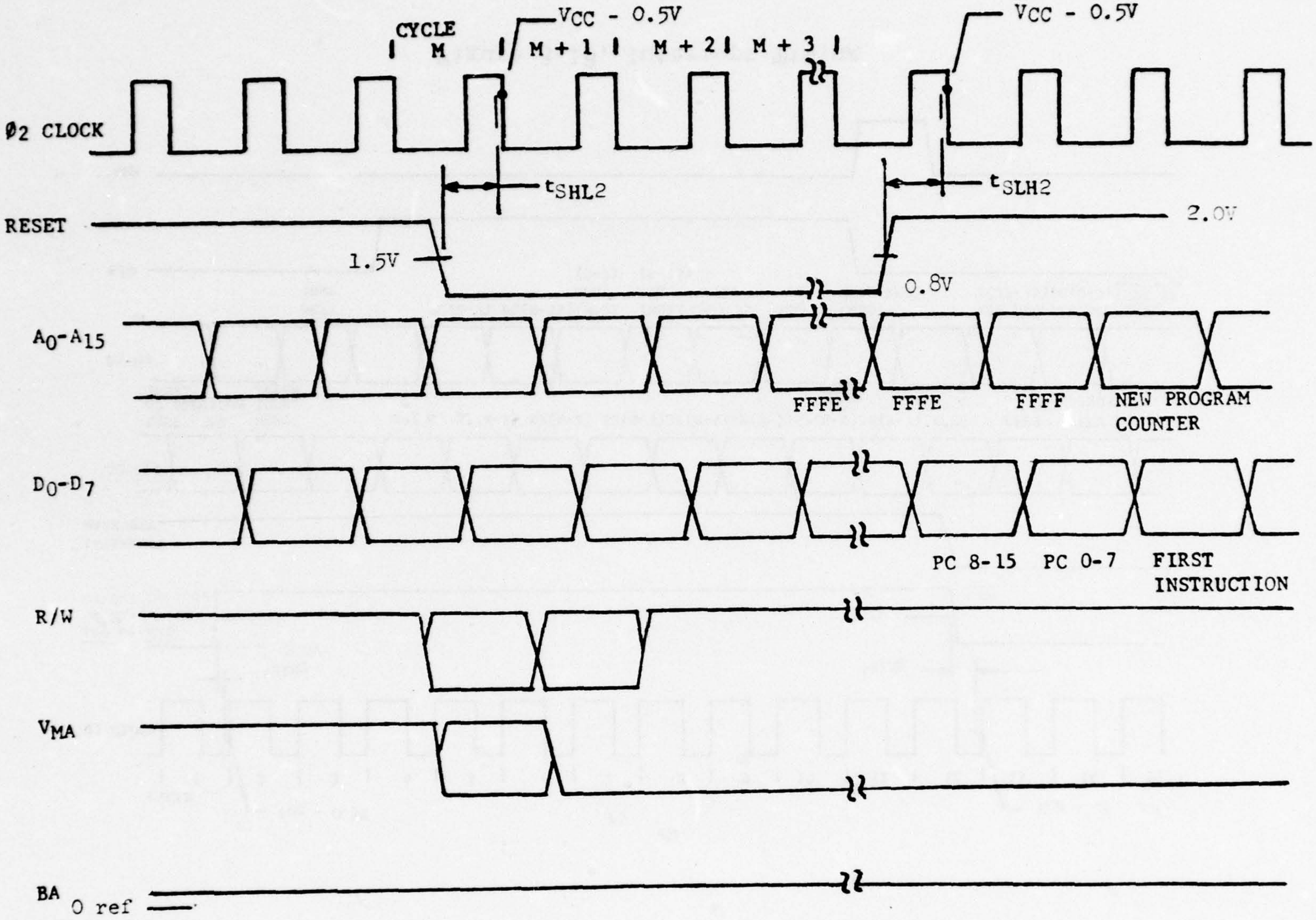


Figure 3-17. Processor Reinitialization-Reset Timing

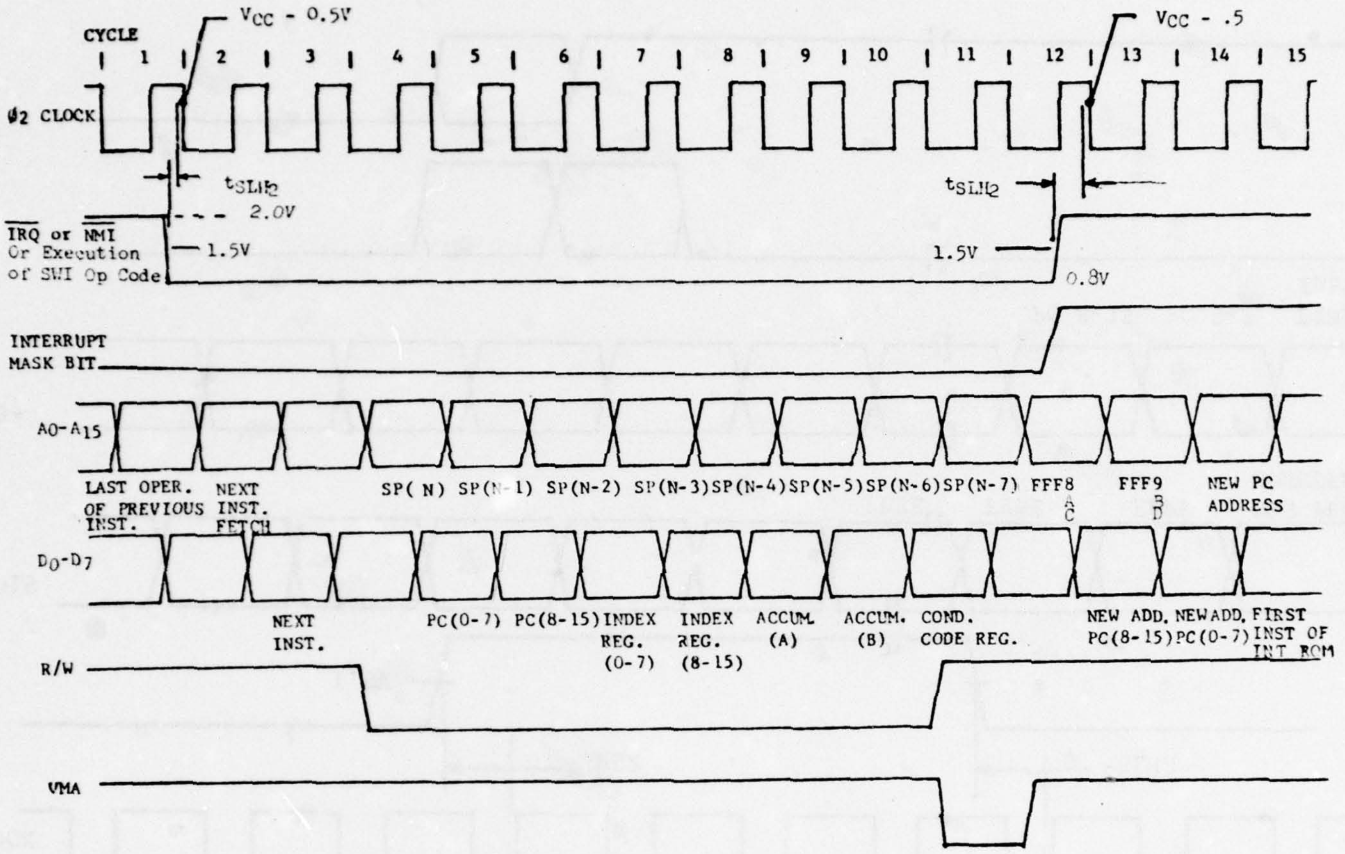
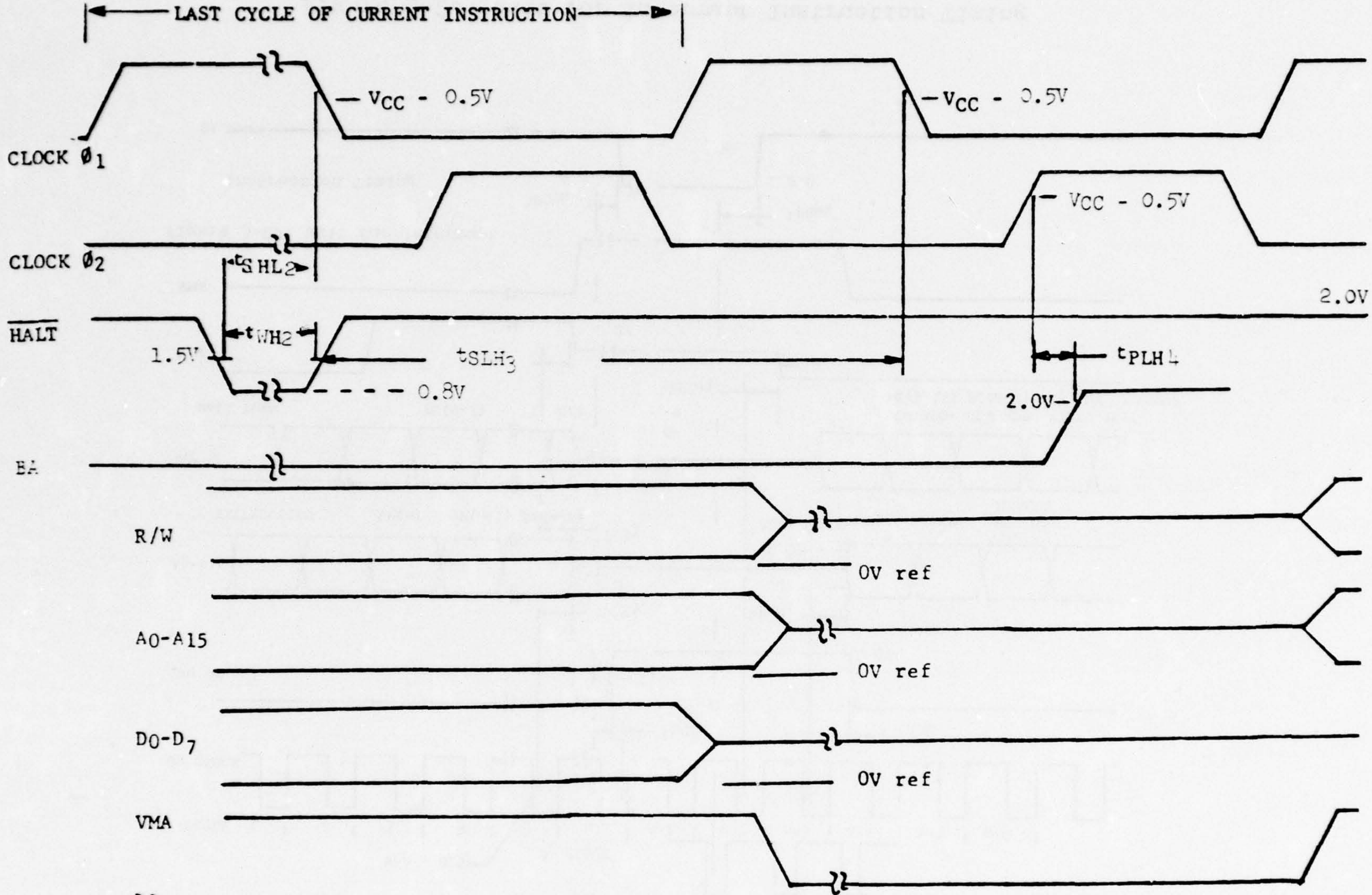


Figure 3-18. Interrupt Timing



≀≀ - can be single or multiple clocks

Figure 3-19. $\overline{\text{Halt}}$ - Cycle Skip

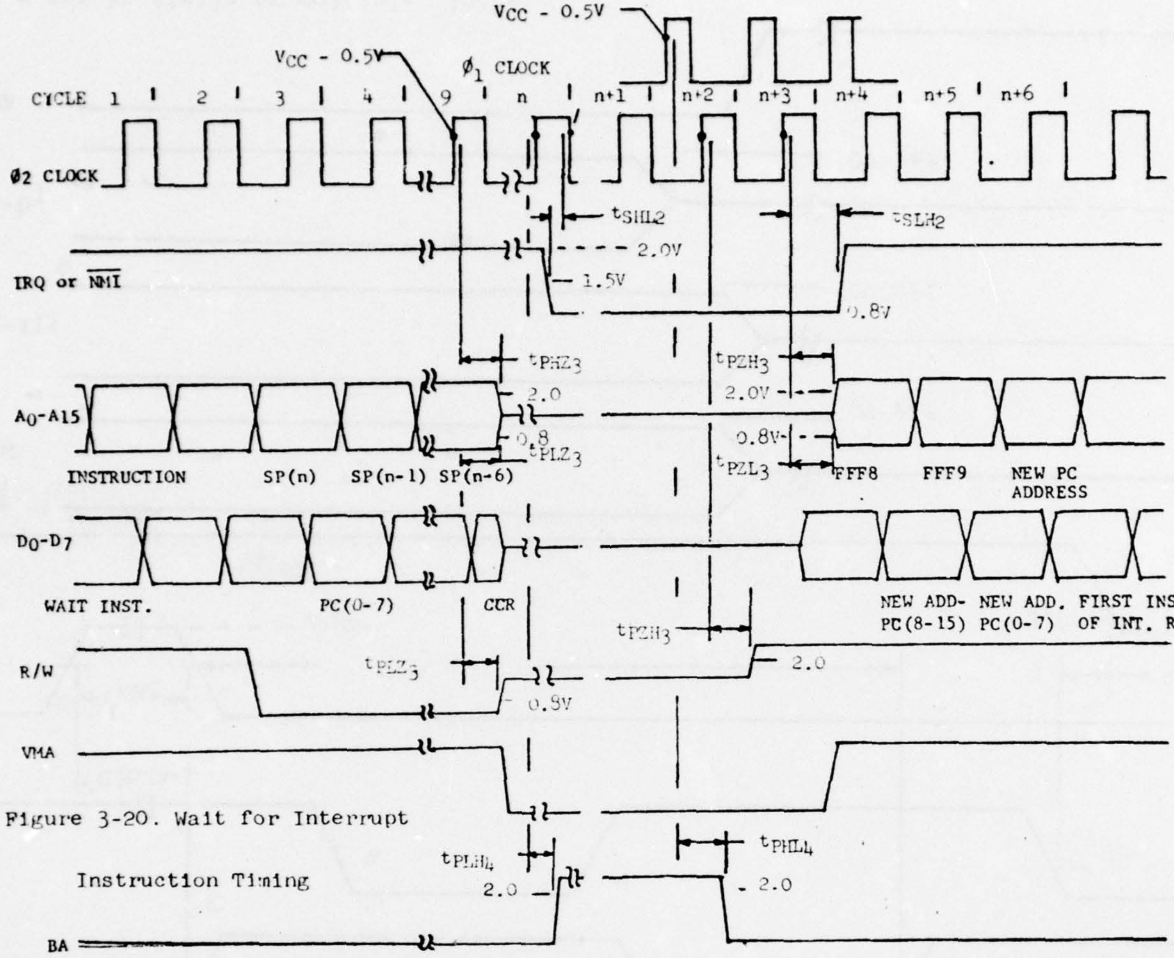


Figure 3-20. Wait for Interrupt Instruction Timing

Figure 3-20. Wait for Interrupt Instruction Timing

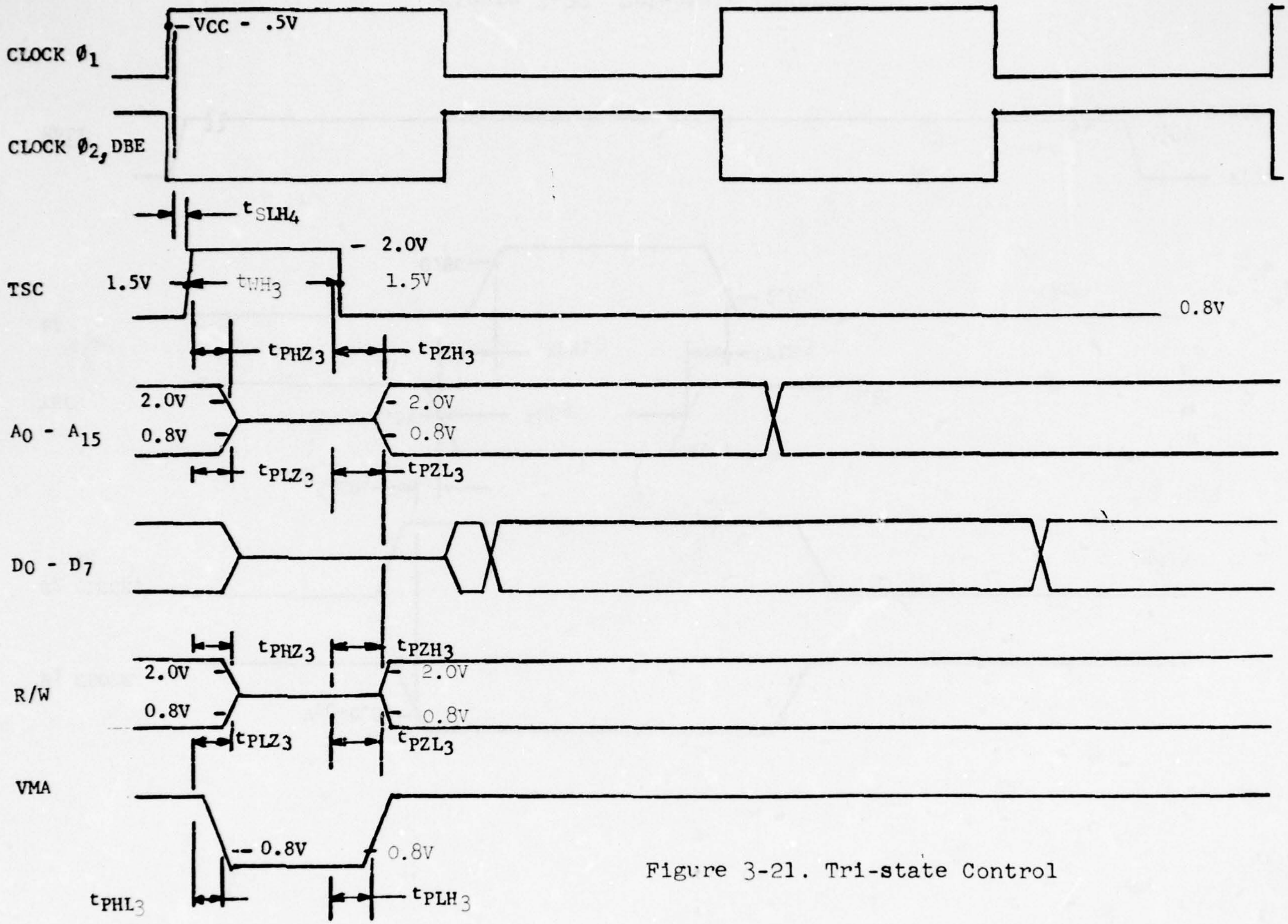


Figure 3-21. Tri-state Control

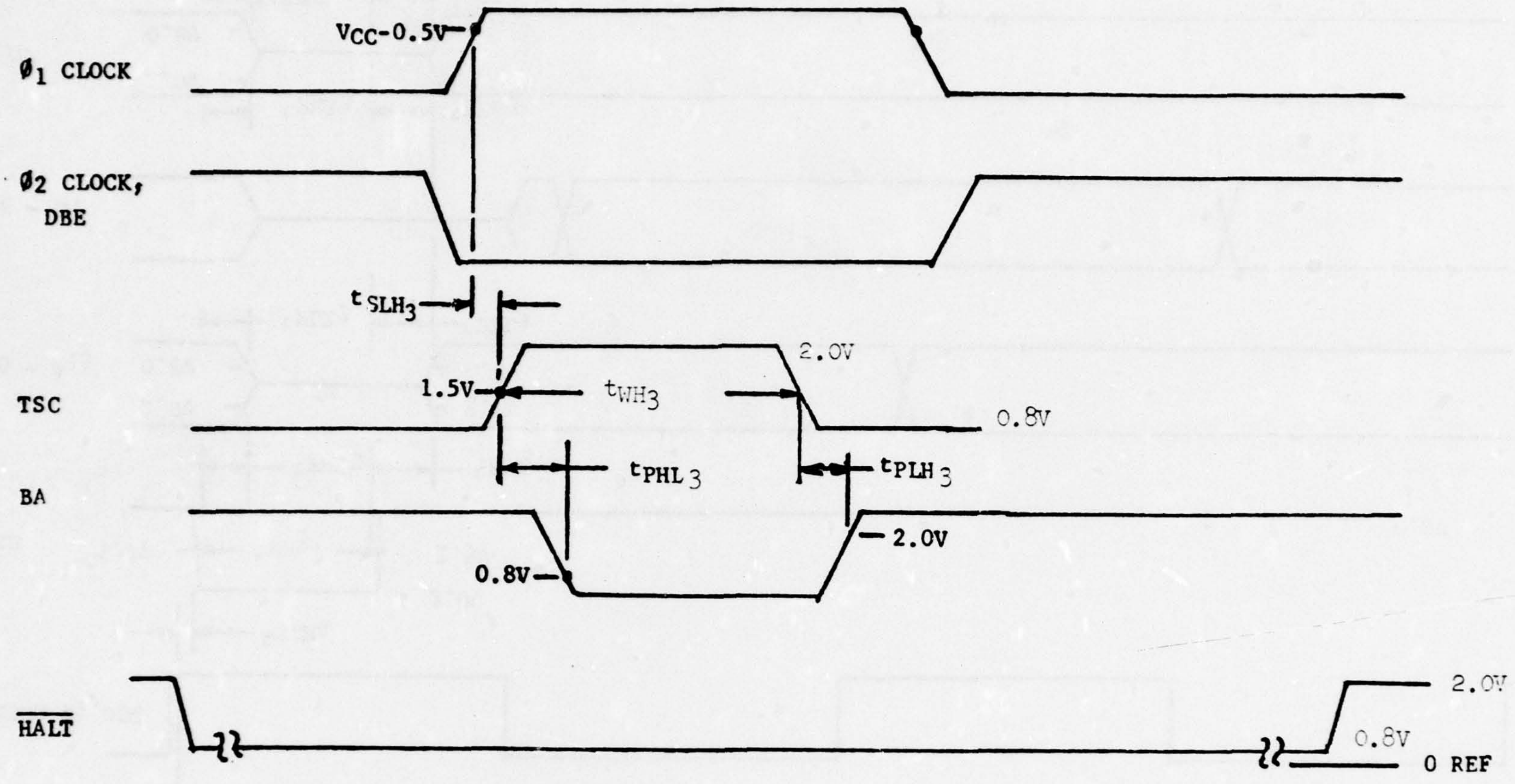


Figure 3-22. Tri-state Control

SECTION IV

TABLE OF CONTENTS

SECTION IV

LIST OF FIGURES

SECTION IV

LIST OF TABLES

SECTION IV

TEST DEVELOPMENT FOR THE VENDOR B I8080A MICROPROCESSOR

4.1 Objective

Review Vendor B's 8080 functional test set, determine its effectiveness and prepare a Logic Integrity Test (LIT) for inclusion in a MIL-M-38510 slash sheet for the 8080.

4.2 Introduction

Vendor B developed a set of vectors which was intended for production testing of the 8080 Microprocessor and for debugging any design errors, particularly in the microprograms (PLA). Since the development of the chip, Vendor B has been examining both in-house and field failures to determine which failures are not caught by the test. Vendor B stated that vectors have been added to discriminate between functional failures and pattern sensitivity failures. This report details:

1) The evaluation of the functional test set to determine its effectiveness with respect to the test criteria detailed in Section III on test development for the Vendor A MC6800 Microprocessor.

2) The development of additional test vectors where required.

3) The specification of waveforms and signal sequencing for applied signals.

The operation of the microprocessor was previously examined and a method of analyzing the test vectors was developed in Section III and Appendices. The approach taken in analyzing the effectiveness of the new LIT is:

1) Analyze the test vectors to determine the rationale used and the completeness of the test based on verification of the functional sectors, data paths and instruction set.

2) Discuss any problems and findings resulting from the evaluation with Vendor B personnel.

3) Determine if the test is a good basis for the slash sheet; if so, specify additional tests as required.

4) Define the timing and waveforms to use for the input and output lines.

4.3 Summary

In order to analyze the new functional test, the vectors were entered into the H-635 computer as the data base used for analysis. This data was "disassembled" to determine the op codes, the addresses, and the data and control signals used. The Vendor B assembler and interpreter programs were then run on these op codes to determine the internal register states and data flow through the ALU.

This functional test is similar to the Vendor A test set, in that it was not developed as a methodical sector by sector test. This, in itself, does not detract from the fault detection capability of the test, but does complicate analysis of the test. This type of test set does not lend itself to fault isolation analysis of a failed device because each vector is apt to be testing previously untested circuitry in each of several sectors scattered over the chip.

There are two distinctly different types of testing performed in this set. In one part the designers made a point of viewing the internal state of the processor frequently. This allowed for an easier examination of this part of the test and helped eliminate the problem of a fault masking itself when faulted circuitry is exercised more than once prior to viewing the intermediate results. The other type of test involved repetitions of a group of instructions. These areas frequently exercised registers and the ALU for long periods without viewing the intermediate results. Vendor B explained that these areas checked for sensitivities which the designers felt might exist and for sensitivities which had appeared in use. Even though these sections of the test did not contribute significantly to the testing confidence based on the test philosophy, they could contribute significantly to the intangible aspects of testing such as instruction and register sensitivities.

The number of vectors in this test is significantly greater (9 to 1) than in the test for the MC6800. There are two reasons for this. First the 8080 requires 3 to 5 clocks per "machine cycle" compared to 1 for the 6800. The second reason is the addition of tests for sensitivity testing.

On a sector by sector basis, (see Figure 4-1) the effectiveness of the test as supplied by Vendor B is as follows:

1) I/O buffers and buses - meets test criteria.

2) Timing and Control circuitry - meets test criteria.

3) Instruction Decode circuitry - The following op codes were not used and/or their operation verified.

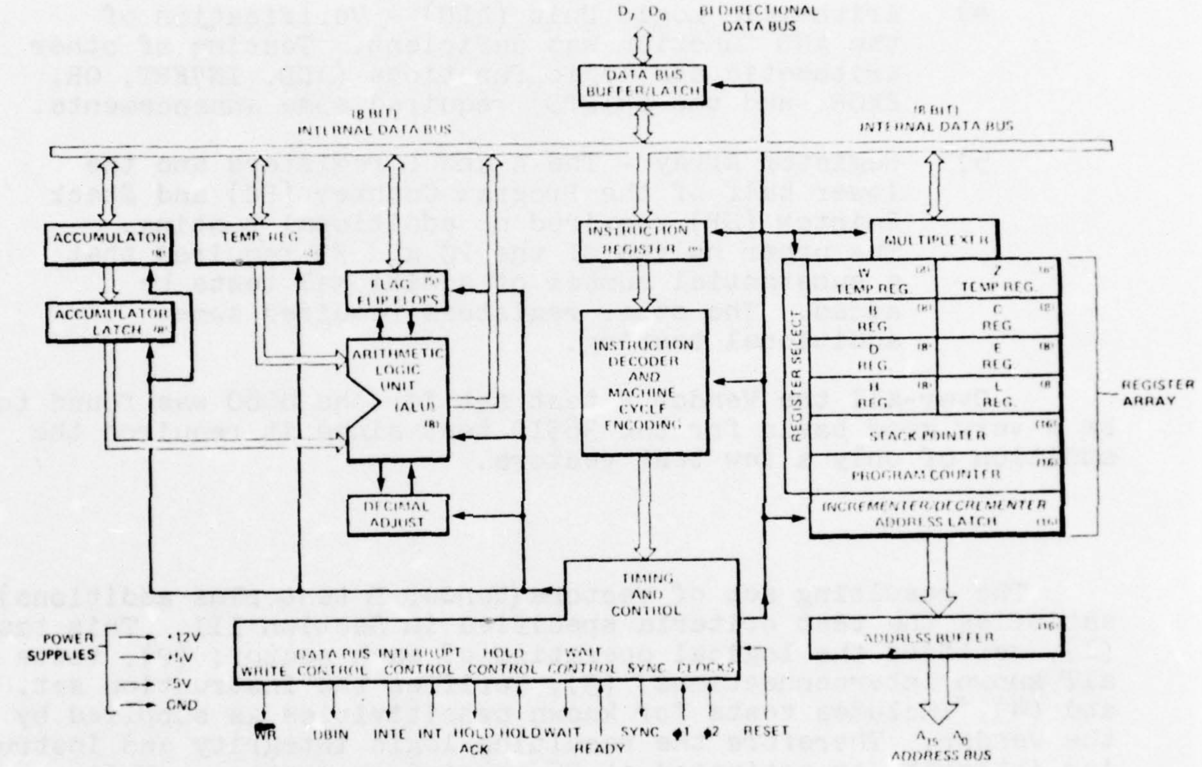


Figure 4-1. Vendor B 8080 Block Diagram

LDA		Load accumulator direct
LHLD		Load H and L direct
ANA	M	AND memory to accumulator
ORA	M	OR memory to accumulator
RST		Restart (certain modes)
DAA		Decimal adjust (certain modes)

All other op codes were used and verified.

4) Arithmetic Logic Unit (ALU) - Verification of the AND function was deficient. Testing of other arithmetic and logic functions (ADD, INVERT, OR, EXOR, and the SHIFTS) required some enhancements.

5) Register Array - The A and L registers and the lower half of the Program Counter (PC) and Stack Pointer (SP) required no additional testing. The upper halves of the PC and PS required that a substantial number of additional tests be added. The other registers required some additional testing.

Over-all the Vendor B test set for the 8080 was found to be a very good basis for the 38510 test since it required the addition of only a few test vectors.

The resulting set of vectors (Vendor B test plus additions) satisfies the test criteria specified in Section III. This test, (1), verifies the logical operation of each sector; (2), tests all known interconnections; (3), verifies the instruction set, and (4), includes tests for known sensitivities as supplied by the vendor. Therefore the resulting logic integrity and instruction integrity is estimated at 95 percent or greater. This assesment is made using similar experience on devices resembling the functional blocks identified within the microprocessor.

Four sets of input timing and voltage conditions were specified which are considered a worst case exercise of the dynamic portions of the uP. By appling the functional test set four times, (one for each of the input specifications), one obtains a high confidence of the integrity of the dynamic buses and dynamic timing circuitry under all input conditions.

4.4 Technical Discussion

4.4.1 Data and Output, Buffers and Buses

The circuitry analyzed in this section includes:

1) The data line buffers and associated chip pins,

2) An 8-bit data bus connecting data buffers, the registers and the ALU,

3) A 16-bit address bus from the program counter, stack pointer, index register and address incrementor/decrementor to the address buffers.

Each flip-flop in the data address buffers was checked for 1 to 1, 1 to 0, 0 to 0 and 0 to 1 transitions. In addition, each data bus line and the associated buffer flip-flops and chip pins were shown to be independent from each other. The buses interconnecting the internal registers with the output buffers were verified as an integral part of the internal registers.

Operation of the data and output buffers was verified. No additional test vectors are required.

4.4.2 Timing and Control

The inputs to the Timing and Control Sector determine the operation of the uP independent of the op codes which are being executed. These inputs are the phase 1 and phase 2 clocks, and the RESET, INTERRUPT, HOLD, and READY inputs.

The clock inputs are inherent to all aspects of the microprocessor's operation. The clock lines are automatically verified if the vector set is sufficient for each functional block, e.g., one of the clocks defines the time when a particular register is loaded with data. If the line is faulted, either the register will not load when so instructed or the register will be loaded even when loading is not specified. Both of these cases will be detected as bad data in the register.

A test of the remaining control lines should check that:

1) Each performs the intended function.

2) The proper priority is maintained when any two or more are applied at the same time.

3) Each will perform independent of the previous instruction.

In the test set each of these control functions is exercised. Operation and verification of these are described below.

The RESET line is used to initialize the processor to a state from which one can begin executing a program. This requires that the Program Counter value be known and that the state of the timing circuitry be known. The RESET sets the Program Counter equal to zero and the timing circuitry to M_1 and T_1. Both of these actions were performed and verified.

The INTERRUPT line in normal operation signals the processor that an activity with a higher priority wants the processor to respond to it rather than to the activity presently engaged in. The uP raises the interrupt acknowledge flag and retains the present value of the program counter during the next instruction. The INTERRUPT can be inhibited by the INTE flip-flop.

The test verified that the INTE flip-flop could be set and reset and that it did mask and enable the INTERRUPT. It also verified the proper results from an interrupt. Therefore the operation of this circuitry was verified.

The HOLD line is used for Direct Memory Access (DMA) applications. A logic high on this line signals the processor to suspend activity and put the Address and Data lines in the high impedance state. The indicator of a hold state is the HLDA (Hold Acknowledge) pin. Verification of the HOLD function must show that the Address and Data lines will float:

1) if the internal HALT flip-flop is set,

2) during T3 if recognized during a read cycle, or

3) after T3 if recognized during a write cycle.

These operations are checked by the test.

The READY is a signal from a memory device, I/O device or other peripheral to indicate when it has stable data ready for the processor or that it has accepted the data from the processor. The response from the processor is the WAIT signal. These lines are exercised and their operation verified during the test whenever the tester is reloading a 1024 vector block (one buffer capacity) and also at various other places.

The interrelations of these functions are verified. Over-all the test set meets the requirements for the Timing and Control Sector.

4.4.3 Instruction Decode Circuitry

A test of the instruction decode circuitry must verify that each instruction correctly controls the necessary sectors and that no additional sectors are affected.

The Vendor B 8080 instruction set defines the operation for 244 of the possible 256 op code combinations (12 are illegal op codes). However, these 244 op code combinations do not represent 244 unique instructions. Several instructions differ only in the selection of a source and/or destination register for a data movement operation. Each of the instructions in one group moves data into one of seven registers. The particular source register for each of these instructions is defined by instruction data bits D_0, D_1 and D_2. This group of instructions can be identified in Figure 4-2 (8080 Instruction Set) by the three "S's" for bit positions D_0, D_1, and D_2. This indicates a mechanism similar to the Vendor A 6800 address modification circuitry, which selects a source register for each of these 8080 instructions. A test for these instructions requires that the operation of each instruction (e.g., MOV M, r) be verified and also that the source register selection circuitry be verified sometime during the verification of these instructions A similar mechanism exists for selecting a destination register for another group of instructions. These are identified in Figure 4-2 by a "D" designation in data bits D_3, D_4, and D_5. A verification of these instructions required that each instruction (e.g., INR, r) be verified and also that the destination register selection circuitry be verified.

In summary the instruction decode circuitry was subdivided into three sectors for analysis. Based on this representation, only four instructions were not verified: LDA (Load Accumulator Direct). LHLD (Load H and L Direct), ANA M (AND Memory to Accumulator) and ORA M (OR Memory to Accumulator). It is possible that operation of these instructions is verified by the test set if further partitioning of the instruction decode sectors can be accomplished. This subdivision is not apparent, so these instructions were added to the test set.

Two other instructions required additional verification; one was the RST (Restart) instruction. The Vendor B 8080 Microcomputer Systems User's Manual shows that this instruction loads data bits D_3, D_4, and D_5 into the corresponding bits of the Z temporary register and 0's into the remaining bits of Z. This is used as the starting address for the next instruction. The test set used and verified RST 1 and RST 5. This means that the path for data bit D_5 into the Z temporary register was verified for operation at 1 and 0 but that D_4 was never verified at 1 (S at 0) and D_3 was never verified at 0 (S at 1). An RST 2 was added to verify the operation of these lines.

Mnemonic	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
		Instruction Code[1]								
MOV $_{r1,r2}$	Move register to register	0	1	D	D	D	S	S	S	5
MOV M, r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r, M	Move memory to register	0	1	D	D	D	1	1	0	7
HLT	Halt	0	1	1	1	0	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
INR r	Increment register	0	0	D	D	D	1	0	0	5
DCR r	Decrement register	0	0	D	D	D	1	0	1	5
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10
CALL	Call unconditional	1	1	0	0	1	1	0	1	17
CC	Call on carry	1	1	0	1	1	1	0	0	11/17
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/17
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17
CP	Call on positive	1	1	1	1	0	1	0	0	11/17
CM	Call on minus	1	1	1	1	1	1	0	0	11/17
CPE	Call on parity even	1	1	1	0	1	1	0	0	11/17
CPO	Call on parity odd	1	1	1	0	0	1	0	0	11/17
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	5/11
RNC	Return on no carry	1	1	0	1	0	0	0	0	5/11

Mnemonic	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
		Instruction Code[1]								
RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
RP	Return on positive	1	1	1	1	0	0	0	0	5/11
RM	Return on minus	1	1	1	1	1	0	0	0	5/11
RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
RST	Restart	1	1	A	A	A	1	1	1	11
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
POP B	Pop register pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	18
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable interrupt	1	1	1	1	0	0	1	1	4
NOP	No operation	0	0	0	0	0	0	0	0	4

NOTES: 1. DDD or SSS – 000 B – 001 C – 010 D – 011 E – 100 H – 101 L – 110 Memory – 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.

Figure 4-2. Instruction Set for the 8080

The other instruction which required additional verification was the DAA (Decimal Adjust) instruction. The operation of this instruction is as follows:

The eight-bit hexadecimal number in the accumulator is adjusted to form two four-bit binary-coded-decimal digits by the following two-step process:

Step 1) If the least significant four bits of the accumulator represent a number greater than 9, or if the carry bit is equal to one, then six is added to the accumulator. Otherwise, no addition occurs.

Step 2) If the most significant four bits of the accumulator now represent a number greater than 9, or if the normal carry bit is equal to one, then 60_{16} is added to the accumulator. Otherwise, no addition occurs.

Note that this instruction is used when adding decimal numbers. It is the only instruction whose operation is affected by the auxiliary carry bit.

Verification of this instruction requires that both steps be verified for the following conditions:

Condition 1) The four appropriate bits greater than 9 and the appropriate carry equal to 0.

Condition 2) The four bits less than 9 and the carry equal to 1.

Condition 3) The four bits less than 9 and the carry equal to 0.

The Vendor B test verified only Step 1/Condition 1 and Step 2/Condition 3. The additional conditions were verified in the final test set.

There were several places in the test set where the designers looped-on (repeated) a group of instructions. The loops contained varying numbers of instructions to view the contents of the registers. Four of the loops consisted of the following group of instructions:

Rotate*		Rotate Accumulator
PUSH	PSW	Push Accumulator and Flags into Stack
MOV	A,M	Move Data from Memory into Accumulator
CMP	B	Compare B with Data
JNZ	$31B_{16}$	Jump if ZERO Flip-flop = 0
DCR	D	Decrement Register D
POP	PSW	Pop Stack into Accumulator and Flags
JMP	$30F_{16}$	Unconditionally Jump

*RLC, RAL, RRC, and RAR for the first, second, third and fourth loops respectively.

These loops put a single "1" in the accumulator and rotate the "1" completely around the register, (8 or 9 rotates depending on the instruction). This loop provides fairly good visibility in that the A, B, D, H, L, SP and the Condition Code Register are examined after each Rotate. Therefore this loop checks for some inter-register disturbances. Also the D Register is checked at the end after 8 decrements from 7 through 0.

After these loops is a very large loop or group of loops (approximately 2500 vectors). This area involves a large amount of data movement within the register array but include very few instructions which give visibility to the contents of the register array. The data starts out as high numbers in all registers. Gradually the numbers are reduced to zeros. Vendor B personnel were contacted to determine their motives in this section. They responded that the section was designed to check for instruction and memory sensitivities. That is, they felt that a good "stuck-at" test had been performed elsewhere and that this long sequence of instructions was performed with checks primarily at the ends of sequences for detection of sensitivities. This section was not changed, since it represents the manufacturer's own ideas on the location of potential sensitivities and checks for known failure mechanisms.

The final test set includes verification of the additional instructions so that all instructions are verified.

4.4.4 Arithmetic Logic Unit

The test philosophy used for an arithmetic adder/subtractor whose mechanization is not known is to apply all possible input combinations to each bit. That is, for the add and subtract modes:

1) Apply all possible inputs (0 & 0, 0 & 1, 1 & 0 and 1 & 1) to each adder input pair with its carry equal to "zero".

2) Apply all possible inputs to each adder input pair with its carry-in a "one".

Two's complement subtraction is performed by complementing the subtrahend and:

1) For single precision arithmetic adding with carry-in equal to "one" or

2) For multiple precision arithmetic adding with carry-in equal to "one" for the first byte and for succeeding bytes, with carry-in equal to "borrow" from preceeding bytes.

Thus a test for subtraction requires verification of the addition function and then verification of the complementing circuitry. The required tests are:

1) Verify the eight possible inputs to each bit of the adder,

2) Verify that the complementing circuitry will complement both a "one" and a "zero" for each bit,

3) Check bit independence and

4) Verify decimal adjust circuitry.

The test philosophy used for the logic operations is to:

1) Apply the following input conditions to each input pair

   0 & 0, 0 &1, 1 & 0 and 1 & 1 during EXOR operations
   0 & 0, 0 & 1, and 1 & 0 during OR operations
   0 & 1, 1 & 0 and 1 & 1 during AND operations

2) Check that for shift left and for shift right operations both "0","1" are shifted from each bit into a "1" and a "0" respectively, in each adjoining bit. Therefore, four shift left and four shift right combinations are required for each bit.

The test set was augmented to detect some additional tests for the OR, EXOR, and ADD functions. The NOT and SHIFT functions were totally verified; the AND required a significant number of additional tests.

The detailed results for each of the ALU modes are in Tables 4-1 through 4-7. The nomenclature used in these tables is as follows:

X - Detected in the Vendor B test.
Blank - Not detected in the Vendor B test.

Input Conditions			Bit Position							
Carry	DI$_1$	DI$_2$	7	6	5	4	3	2	1	0
0	0	0	X	X	X	X	X	X	X	X
0	0	1	X	X	X	X	X	X	X	
0	1	0	X	X		X	X	X	X	X
0	1	1	X	X	X		X	X	X	X
1	0	0	X	X				X	X	
1	0	1	X	X	X	X	X	X	X	X
1	1	0	X	X	X	X	X	X	X	X
1	1	1								

Carry Function	Value	
Carry In Add	0	X
	1	X
Carry In Subtract	0	X
	1	X
Carry Out	0	X
	1	X

Table 4-1. Conditions Applied to ALU in ADD (SUBTRACT) Mode

Input Condition		Bit Position							
DI$_1$	DI$_2$	7	6	5	4	3	2	1	0
0	0	X					X		
0	1		X		X	X		X	X
1	0	X	X	X	X	X	X	X	X

Table 4-2. Conditions Applied to ALU in OR Mode

Input Condition		Bit Position							
DI$_1$	DI$_2$	7	6	5	4	3	2	1	0
0	1								
1	0							X	X
1	1	X		X			X		

Table 4-3. Conditions Applied to ALU in AND Mode

Input Condition		Bit Position							
DI$_1$	DI$_2$	7	6	5	4	3	2	1	0
0	0	X	X		X	X	X	X	X
0	1	X	X	X					
1	0			X	X	X	X		
1	1	X	X	X	X	X	X	X	

Table 4-4. Conditions Applied to ALU in Exclusive-OR Mode

Input Condition	Bit Position							
	7	6	5	4	3	2	1	0
0	X	X	X	X	X	X	X	X
1	X	X	X	X	X	X	X	X

Table 4-5. Conditions Applied to the One's Complementor

Input Condition	Bit Position	Carry In		Carry Out	
	7 6 5 4 3 2 1 0	From Carry F/F	From Bit 0	To Carry F/F	To Bit 7
0	X X X X X X X X	X	X	X	X
1	X X X X X X X X	X	X	X	X

Table 4-6. Integrity of the RIGHT Shift Circuitry

Input Condition	Bit Position	Carry In		Carry Out	
	7 6 5 4 3 2 1 0	From Carry F/F	From Bit 7	To Carry F/F	To Bit 0
0	X X X X X X X X	X	X	X	X
1	X X X X X X X X	X	X	X	X

Table 4-7. Integrity of the LEFT Shift Circuitry

4.4.5 Register Array

The designers of the Vendor B 8080 deliberately used a large area of the chip for the registers so that the pattern and temperature sensitivities associated with RAMS would be eliminated Therefore, the evaluation criteria for the register array was to check for:

1) register independence,

2) bit independence, and

3) flip-flop integrity (i.e., insure transitions of 0 to 0, 0 to 1, 1 to 1 and 1 to 0 for each bit, each register).

For those testers with algorithmic hardware pattern generators, walking 1/0 and galloping 1/0 patterns could be applied to the register arrays with relatively few test instructions. Since a truth table representation of these patterns is prohibitively large and Vendor B asserts that little is gained by these patterns, they are not included nor required as part of the test set. If it is later determined that this type of pattern detects a significant number of faulty chips which are otherwise passed by t he 38510's test, then the patterns will be added at that time.

In general the test set verified the operation of most of the register array. Several areas required no additional tests: the A and L Registers and the lower half of the Stack Pointer and Program Counter Registers. The registers requiring the most additional tests were the upper bits of the Stack Pointer and the Program Counter. Many of these bits were never loaded with a 1. As such these bits were never tested in the original test set for integrity and independence.

The INSTRUCTION Register was not analyzed as part of the register array but is included as an integral part of the Instruction Decode Circuitry; i.e., The instruction register is verified by insuring that all instructions are executed correctly. The remaining registers (B,C,D,E, and H) required that some additional tests be performed.

Tables 4-8 through 4-13 detail the faults detected in the register array by the Vendor B test set. Tests were generated to catch the remaining faults and were added to the final test set.

The nomenclature used in the tables is as follows:

X - Detected using Vendor B's vectors
Blank - Not detected by Vendor B vectors
. - Not possible as a bit cannot be independent from itself

	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X
0 0	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X
1 0	X	X	X	X	X	X	X	X
1 1	X	X	X	X	X	X	X	X

BIT AT ZERO

BIT AT ONE								
0	X	X	X	X	X	X	X	.
1	X	X	X	X	X	X	.	X
2	X	X	X	X	X	.	X	X
3	X	X	X	X	.	X	X	X
4	X	X	X	.	X	X	X	X
5	X	X	.	X	X	X	X	X
6	X	.	X	X	X	X	X	X
7	.	X	X	X	X	X	X	X

A Register Integrity
(a)

	S	Z		AC		P		C
DATA = 0	X	X		X		X		X
DATA = 1	X	X		X		X		X
0 0		X						X
0 1	X	X				X		X
1 0	X	X				X		X
1 1								

BIT AT ZERO

BIT AT ONE								
C	X	X		X		X		.
		X		X			.	X
P						.		
					.			
AC	X	X		.		X		X
			.					
Z	X	.		X				X
S	.	X		X		X		X

Condition Code Integrity
(b)

Table 4-8. A and Condition Code Register Integrity

	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X
0 0	X	X	X		X	X	X	X
0 1	X		X	X	X		X	X
1 0						X	X	X
1 1	X	X	X	X	X	X	X	X

BIT AT ZERO

BIT AT ONE								
0		X			X	X	X	.
1	X	X	X		X	X	.	X
2						.	X	X
3		X			.	X	X	X
4	X	X	X	.	X	X	X	X
5		X	.		X	X	X	X
6		.			X	X	X	X
7	.	X	X		X	X	X	X

B Register Integrity
(a)

	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X
0 0	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X
1 0				X	X	X	X	X
1 1	X	X	X	X			X	

BIT AT ZERO

BIT AT ONE								
0								.
1	X	X	X	X	X	X	.	X
2			X	X	X	.	X	X
3			X	X	.		X	X
4		X		.	X	X	X	X
5		X	.		X	X	X	X
6		.		X	X	X	X	X
7	.	X	X	X	X	X	X	X

C Register Integrity
(b)

Table 4-9. B and C Register Integrity

		7	6	5	4	3	2	1	0
DATA = 0		X	X	X	X	X	X	X	X
DATA = 1		X	X	X	X	X	X	X	X
0	0	X	X	X	X	X	X	X	X
0	1	X	X	X	X	X	X	X	X
1	0	X	X	X	X	X	X	X	X
1	1	X	X	X	X		X	X	
		BIT AT ZERO							
BIT AT ONE	0	X	X	X	X	X	X	X	.
	1	X	X	X	X	X	X	.	X
	2	X	X	X	X	X	.	X	X
	3	X	X	X	X	.		X	
	4	X	X	X	.	X		X	X
	5	X	X	.	X	X	X	X	X
	6	X	.	X	X	X	X	X	X
	7	.	X	X	X	X	X	X	X

D Register Integrity
(a)

		7	6	5	4	3	2	1	0
DATA = 0		X	X	X	X	X	X	X	X
DATA = 1		X	X	X	X	X	X	X	X
0	0	X	X	X	X	X		X	X
0	1	X	X	X	X	X	X	X	X
1	0	X	X	X	X	X	X	X	X
1	1	X	X	X	X		X	X	X
		BIT AT ZERO							
BIT AT ONE	0	X	X	X	X	X	X	X	.
	1	X	X	X	X	X	X	.	X
	2	X	X	X	X	X	.	X	X
	3	X	X			.		X	X
	4	X	X		.	X		X	X
	5	X	X	.		X		X	X
	6	X	.			X		X	X
	7	.	X	X	X	X	X	X	X

E Register Integrity
(b)

Table 4-10. D and E Register Integrity

	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X
0 0	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X
1 0	X	X	X	X	X	X	X	X
1 1	X	X	X	X	X	X	X	X

BIT AT ZERO

		7	6	5	4	3	2	1	0
	0	X	X	X	X	X	X	X	.
BIT	1	X	X	X	X	X	X	.	X
	2	X	X	X	X	X	.	X	X
AT	3	X	X	X	X	.	X	X	X
	4				.	X	X	X	X
ONE	5			.	X	X	X	X	X
	6	X	.	X	X	X	X	X	X
	7	.			X	X	X	X	X

H Register Integrity
(a)

	7	6	5	4	3	2	1	0
DATA = 0	X	X	X	X	X	X	X	X
DATA = 1	X	X	X	X	X	X	X	X
0 0	X	X	X	X	X	X	X	X
0 1	X	X	X	X	X	X	X	X
1 0	X	X	X	X	X	X	X	X
1 1	X	X	X	X	X	X	X	X

BIT AT ZERO

		7	6	5	4	3	2	1	0
	0	X	X	X	X	X	X	X	.
BIT	1	X	X	X	X	X	X	.	X
	2	X	X	X	X	X	.	X	X
AT	3	X	X	X	X	.	X	X	X
	4	X	X	X	.	X	X	X	X
ONE	5	X	X	.	X	X	X	X	X
	6	X	.	X	X	X	X	X	X
	7	.	X	X	X	X	X	X	X

L Register Integrity
(b)

Table 4-11. H and L Register Integrity

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA = 0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DATA = 1							X	X	X	X	X	X	X	X	X	X	X
TRANSITIONS	0→0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	0→1						X	X	X	X	X	X	X	X	X	X	X
	1→0						X	X	X	X	X	X	X	X	X	X	X
	0→1						X	X	X	X	X	X	X	X	X	X	X

		BIT AT ZERO															
BIT AT ONE	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	.
	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	.	X
	2	X	X	X	X	X	X	X	X	X	X	X	X	X	.	X	X
	3	X	X	X	X	X	X	X	X	X	X	X	X	.	X	X	X
	4	X	X	X	X	X	X	X	X	X	X	X	.	X	X	X	X
	5	X	X	X	X	X	X	X	X	X	X	.	X	X	X	X	X
	6	X	X	X	X	X	X	X	X	X	.	X	X	X	X	X	X
	7	X	X	X	X	X	X	X	X	.	X	X	X	X	X	X	X
	8	X	X	X	X	X	X	X	.	X	X	X	X	X	X	X	X
	9	X	X	X	X	X	X	.	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	.	X	X	X	X	X	X	X	X	X	X
	11					.											
	12				.												
	13																
	14		.														
	15	.															

Table 4-12. Program Counter Integrity

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA = 0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DATA = 1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
TRANSITIONS	0→0								X	X	X	X	X	X	X	X	X
	0→1					X	X	X	X	X	X	X	X	X	X	X	X
	1→0					X	X	X	X	X	X	X	X	X	X	X	X
	0→1												X	X	X	X	X

BIT AT ONE		BIT AT ZERO															
	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	.
	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	.	X
	2	X	X	X	X	X	X	X	X	X	X	X	X	X	.	X	X
	3	X	X	X	X	X	X	X	X	X	X	X	X	.	X	X	X
	4	X	X	X	X	X	X	X	X	X	X	X	.	X	X	X	X
	5	X	X	X	X	X	X	X	X			.	X	X	X	X	X
	6	X	X	X	X	X	X	X	X		.		X	X	X	X	X
	7	X	X	X	X	X	X	X	X	.	X	X	X	X	X	X	X
	8	X	X	X	X	X			.	X	X	X	X	X	X	X	X
	9	X	X	X	X	X		.	X	X	X	X	X	X	X	X	X
	10	X	X	X	X	X	.		X	X	X	X	X	X	X	X	X
	11	X	X	X	X	.			X	X	X	X	X	X	X	X	X
	12				.					X	X	X	X	X	X	X	X
	13			.						X	X	X	X	X	X	X	X
	14		.							X	X	X	X	X	X	X	X
	15	.								X	X	X	X	X	X	X	X

Table 4-13. Stack Pointer Integrity

4.4.6 Input and Output Signal Specifications

To obtain a good test of the uP's dynamic parameters, it is necessary to test the uP at clock rates, since the majority of its circuitry is buried and has limited access from the device pins. The large number of test vectors and requirement for precise input conditions necessitates a versatile, high-speed tester. It is practical, and in many cases necessary, to combine input threshold, output level, dynamic and switching tests with the functional test.

Four sets of input conditions were chosen to dynamically test the uP. They are specified in Table 4-14 and graphically shown in Figures 4-3 through 4-13. Each column in Table 4-14 (Conditions A through D) specifies the input conditions to be applied during a pass through the functional test set.

The input voltage levels are specified with V_{IH} equal to the minimum (9.0V for the clocks and 3.3V for all others) and with V_{IL} equal to the maximum (0.8V for all inputs). The three power supplies (VBB, VCC, and VBB) are specified at the most negative voltages (-5%) for the first three passes of the functional test. This ensures that charge transfer circuits operate correctly with the worst case (least amount of) charge. The last pass is specified at the most positive voltages as a verification of operation at worst case input thresholds.

The waveforms and the setup and hold times for the inputs are specified as follows:

1) Period and Clock Specifications - The microprocessor is a cyclical device and, as such, most of the inputs and all outputs are referenced to certain repeating waveforms during each cycle. Each repetition of these signals (e.g., Ø1 and Ø2 clocks) defines the cycle time and also defines the period of each vector (line entry in the functional test). Therefore, the strobe points for these signals are also repeating at the same rate as the clocks. The rate of repetition is referred to as the period. 480 ns and 2 uS were chosen as two periods for the input conditions because they represent the highest (2.08 MHz) and lowest (500 KHz) operating frequencies for the 8080. The first ensures that the internal propagation delays permit maximum frequency operation and the second ensures that leakage in the dynamic portions of the device is low enough such that the uP will run at minimum frequency.

Four sets of input timing and voltage conditions are specified in Table 4-14 and Figure 4-3, each representing a pass through the functional test.

- Worst case conditions are specified for high speed operation during the first pass by specifying the power supplies and clock waveforms as follows. The three power supplies are specified at their most negative limit (-5%). Minimum Ø1 width allows the minimum precharge time on the data bus, and the minimum Ø2 width coupled with minimum phase lag from Ø1 to Ø2 results in the minimum time to transfer data.

- The second pass is similar to the first except that the maximum cycle time is specified. This condition results in the minimum precharge time and the maximum transfer time to the static registers.

- In pass three write time to the dynamic areas is worst cased by widening Ø2 to its maximum width. Otherwise the input conditions are the same as in the second pass.

- The fourth set of input conditions tests for the high input thresholds by raising the three power supplies to their most positive values (+5%). Minimum Ø1 to Ø2 delay is also specified.

2) Data Bus (Input Mode) - On read cycles, the data must be held stable from 150 ns prior to, until 50 ns or t_{p3} after the trailing edge of Ø2 and also 30 ns prior to the trailing edge of Ø1 (see Figure 4-4). The data inputs are specified as driven and stable at the desired logic levels during every read cycle and also specified as driven to the opposite (false) value before and after the valid data-in period during every read cycle. This operation is refered to as a "return to complement" waveform. During wait states (of read cycles) the processor is insensitive to any changes occurring prior to the setup for T_3. This is verified by the Vendor B test by applying several data patterns during a fetch instruction wait. If latching were to occur in a defective 8080, then an erroneous op code would be executed.

AD-A038 969 GENERAL ELECTRIC CO PITTSFIELD MASS ORDNANCF SYSTEMS F/G 9/5
DIGITAL MICROCIRCUIT CHARACTERIZATION AND SPECIFICATION.(U)
MAR 77 T M OSTROWSKI F30602-74-C-0159

UNCLASSIFIED RADC-TR-77-91 NL

2 OF 3
AD
A038969

3) INT - The interrupt line is asynchronous and a request may therefore originate at any time during any instruction cycle. Internal logic re-clocks the external request so that the proper internal timing relationship is established. The application and removal times are specified such that the minimum setup and hold times are defined for both logic 1 and logic 0. Figure 4-9 shows the operation of the processor when an interrupt occurs and graphically shows the setup and hold times by picturing the INT signal as a return to complement signal. Some high-speed test systems cannot generate this type of signal exactly as shown. For those testers it is still necessary to verify both operating modes for each of these signals somewhere within the test using equivalent waveforms.

4) HOLD - HOLD is used to suspend the normal operations of the CPU so that another device can take control of the address and data lines. Like the interrupt, the HOLD line is synchronized internally. The signal is specified similar to INT. That is, the waveform as described by Figures 4-7 and 4-8 are a return to complement with the true data occurring during the critical internal synchronization period. Figure 4-7 is used when the test specifies a "0" for HOLD and Figure 4-8 when a "1" is specified. The same tester restrictions apply to this signal.

5) READY - The READY line is used by memory or I/O devices to signal the processor that it has received data or is ready with data. If peripheral chips are accessed by the processor but are not ready, then the peripheral must signal the processor with a low on READY prior to the trailing edge of Ø2 during T2. When it is ready, it raises READY and the processor will reqognize it during the next cycle provided that READY occurs prior to Ø2. In Figures 4-5 and 4-6, this signal is shown with a return-to-complement waveform.

6) RESET - No setup or hold times are specified by the manufacturer for this signal. These remain to be determined for the slash sheet. The operation of this signal is shown is Figure 4-13.

Table 4-14. Input Conditions
for the Functional Test Set

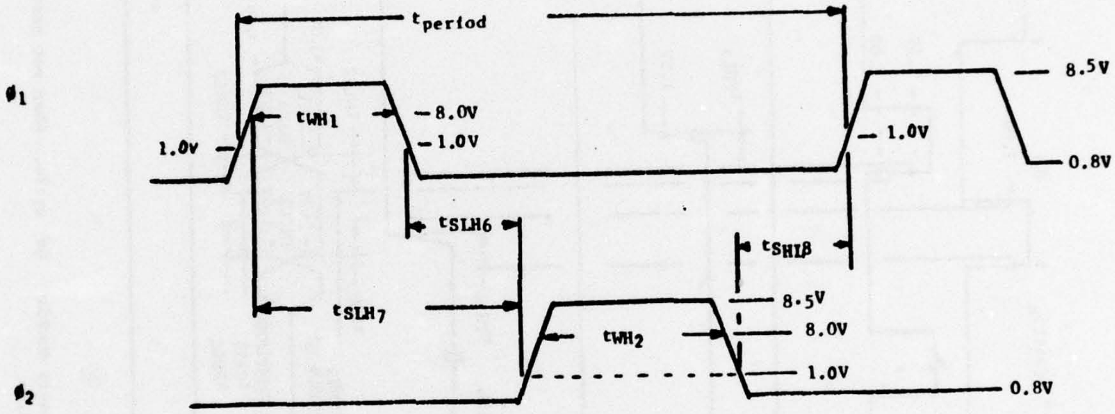
		INPUT CONDITIONS					
SYMBOL	TERMINALS	A	B	C	D	UNITS	FIGURE
V_{BB}	V_{BB}	-5.50	-5.50	-5.50	-4.50	Volts	-
V_{CC}	V_{CC}	+4.50	+4.50	+4.50	+5.50	Volts	-
V_{DD}	V_{DD}	+10.8	+10.8	+10.8	+13.2	Volts	-
V_{IH1}	All logic inputs except clock	3.0	3.0	3.0	3.0	Volts	-
V_{IH2}	clock inputs	8.5	8.5	8.5	8.5	Volts	-
V_{IL1}	All logic inputs including clock	0.8	0.8	0.8	0.8	Volts	-
t_{period}	ϕ_1	0.48	2.0	2.0	.48	us	4-3
t_{WH1}	ϕ_1	80	80	80	60	ns	4-3
t_{WH2}	ϕ_2	220	220	1840	220	ns	4-3
t_{SHL6}	ϕ_2	0	0	0	120	ns	4-3
t_{SLH7}	ϕ_2	80	80	80	180	ns	4-3
t_{SLH8}	ϕ_2	180	1700	80	80	ns	4-3
t_{SLH2}	D0-D_7	130	130	130	30	ns	4-4
t_{SHL2}	D_0-D_7	130	130	130	30	ns	4-4
t_{SLH3}	D_0-D_7	130	130	130	150	ns	4-4
t_{SHL3}	D_0-D_7	130	130	130	150	ns	4-4

(continued on next page)

Table 4-14. Input Conditions for the Functional Test Set

(concluded)

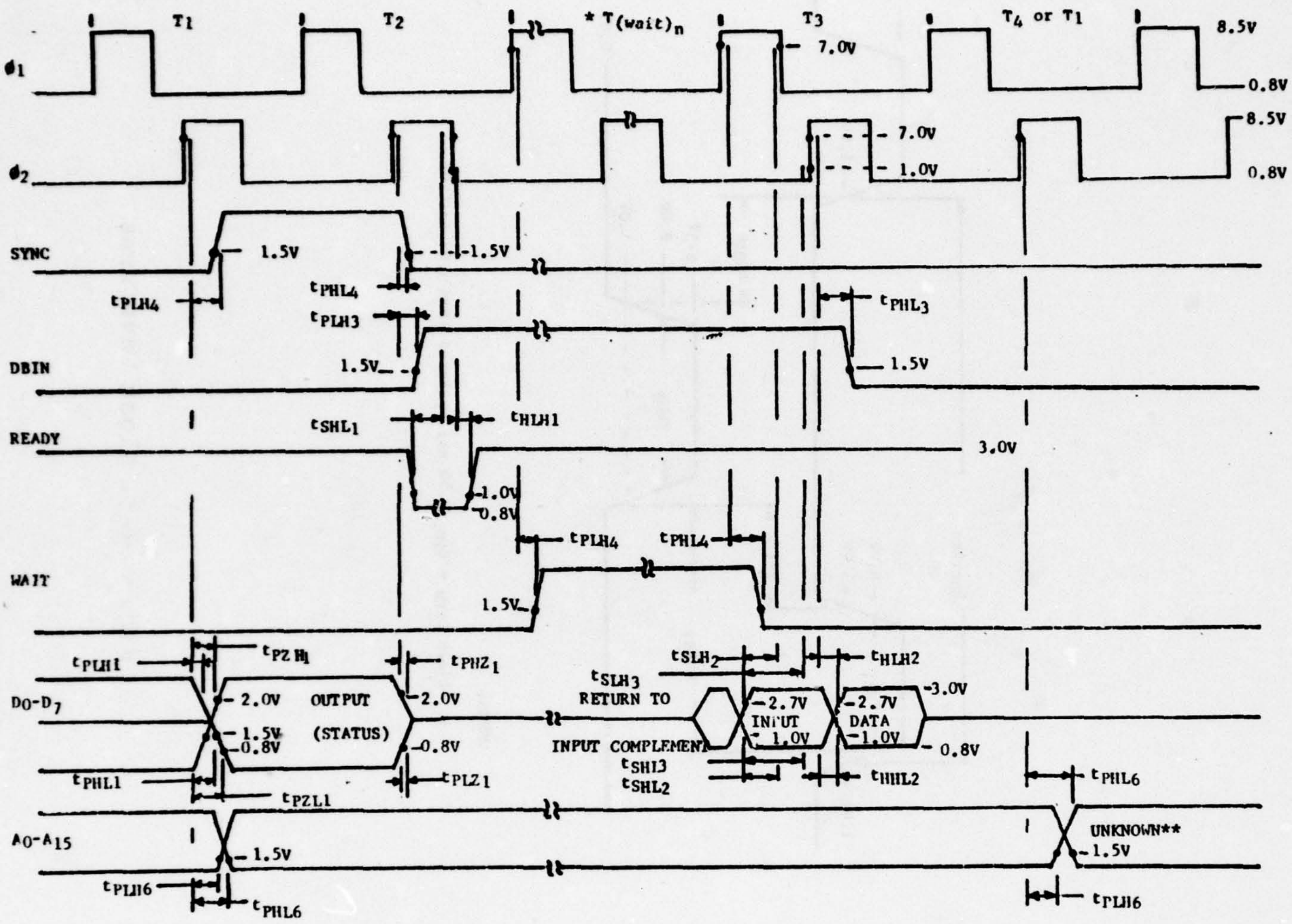
		INPUT CONDITIONS					
SYMBOL	TERMINALS	A	B	C	D	UNITS	FIGURE
t_{HHL2}	D_0-D_7	50	50	50	50	ns	4-4
t_{HLH2}	D_0-D_7	50	50	50	50	ns	4-4
t_{SLH1}	Ready	120	120	120	120	ns	4-6
t_{SHL1}	Ready	120	120	120	120	ns	4-4, 4-5
t_{HHL1}	Ready INT Hold	0	0	0	0	ns	4-6, 4-8 thru 4-10
t_{HLH1}	Ready INT Hold	0	0	0	0	ns	4-4, 4-5, 4-7, 4-9, 4-10
t_{SLH4}	INT	120	120	120	120	ns	4-9, 4-10
t_{SHL4}	INT	120	120	120	120	ns	4-9, 4-10
t_{SHL5}	HOLD	140	140	140	140	ns	4-7
t_{SLH5}	HOLD	140	140	140	140	ns	4-8
t_{SLH9}	RESET	TO BE SUPPLIED BY MANUFACTURER					4-13
t_{HHL3}	RESET	TO BE SUPPLIED BY MANUFACTURER					4-13



NOTES:

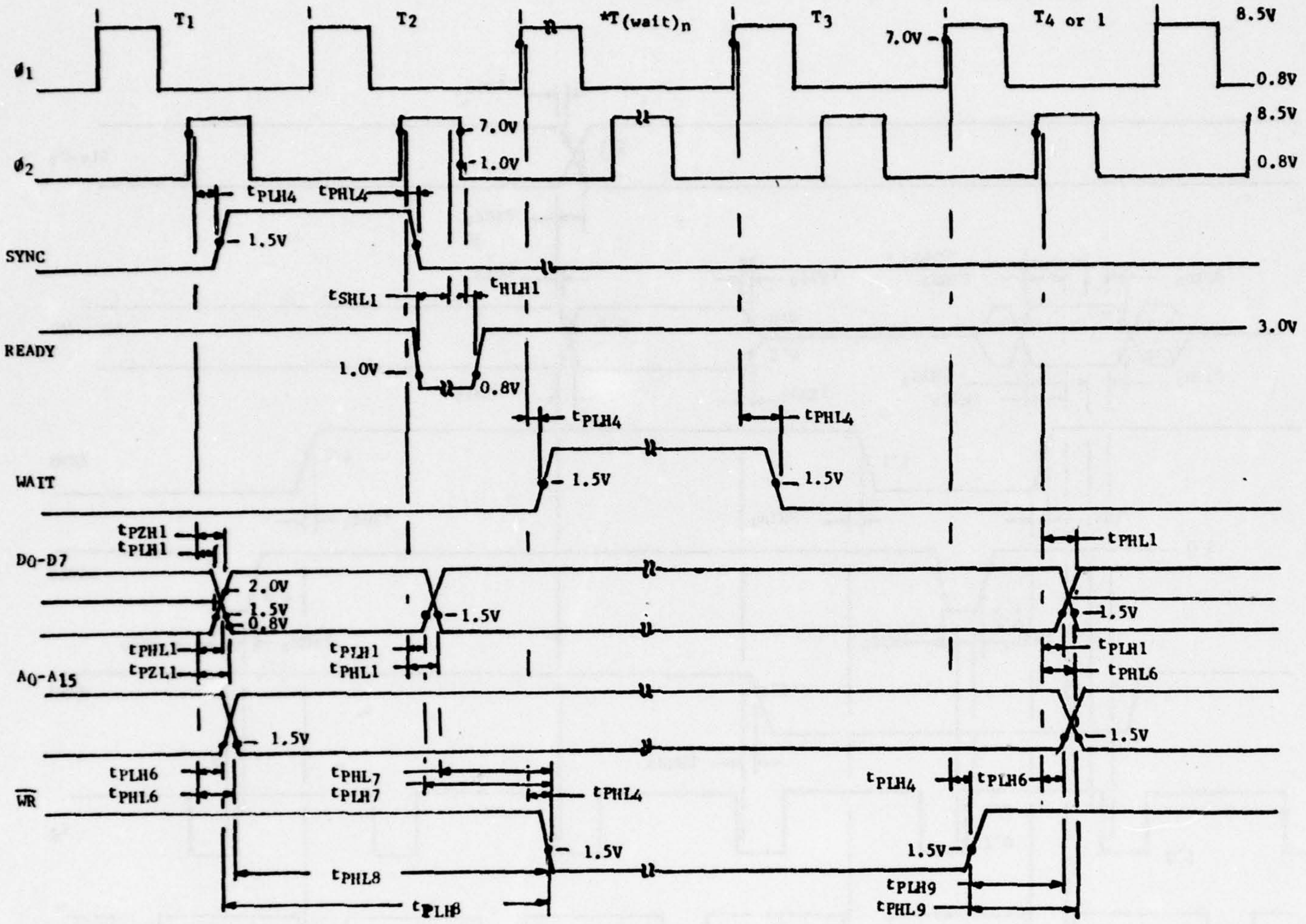
1. Clock $t_{TLH} = t_{THL}$ = 50 ns; measured from 1.0V to 7.0V.

Figure 4-3. Clock Waveforms



*It is not intended that the wait state be entered for every machine (m) cycle. When not performed omit the T wait cycle and disregard the "ready" and "wait" signals.
**Unknown for T4 cycle.

Figure 4-4. READ Cycle with Single or Multiple WAIT Cycles



*It is not intended that the wait state be entered for every machine cycle (T_1-T_4 or T_1-T3). When not performed, omit T(wait) cycle and disregard the "READY" and "WAIT" signals.

Figure 4-5. WRITE Cycle with Single or Multiple WAIT Cycles

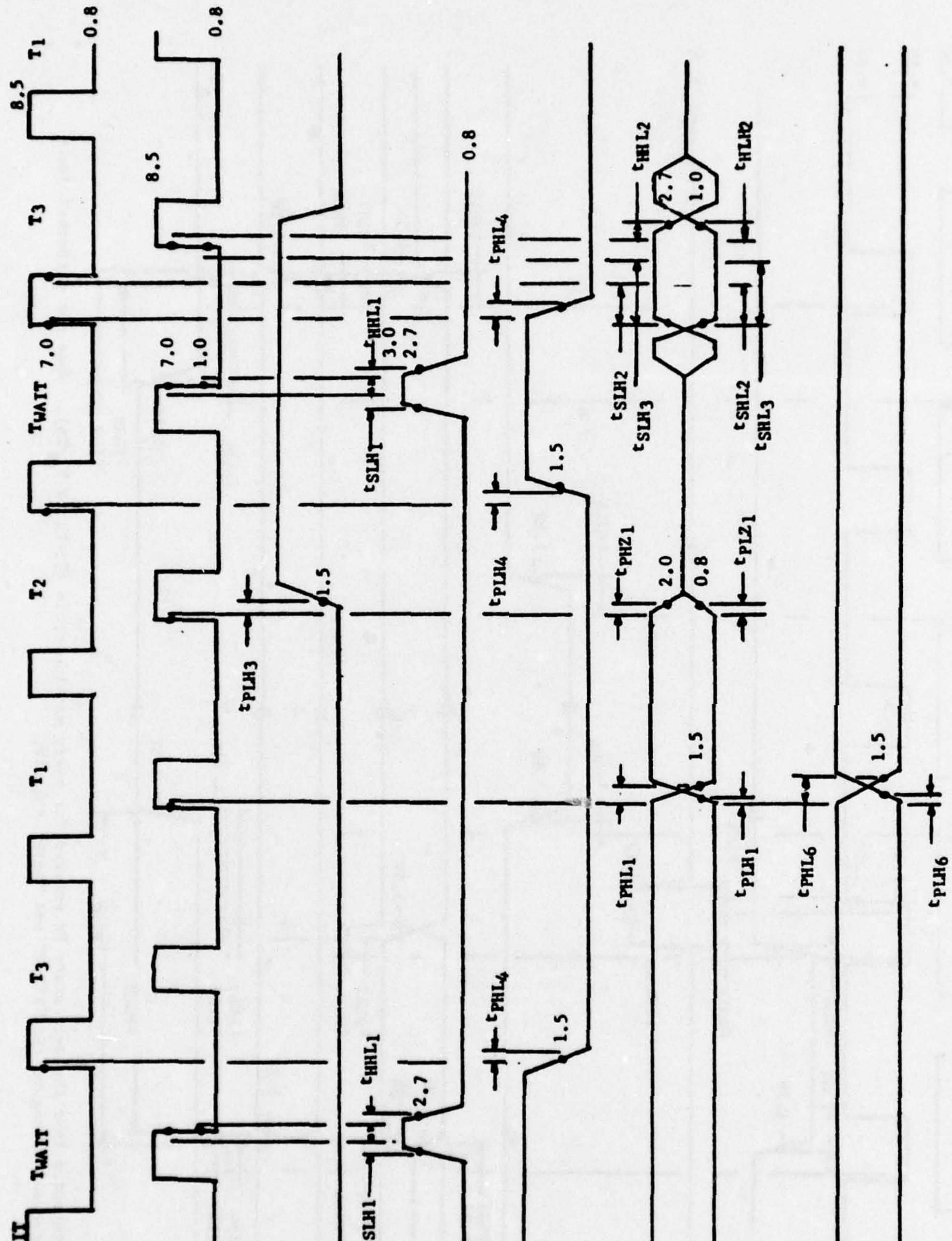


Figure 4-6. WAIT - Machine Cycle EXECUTE

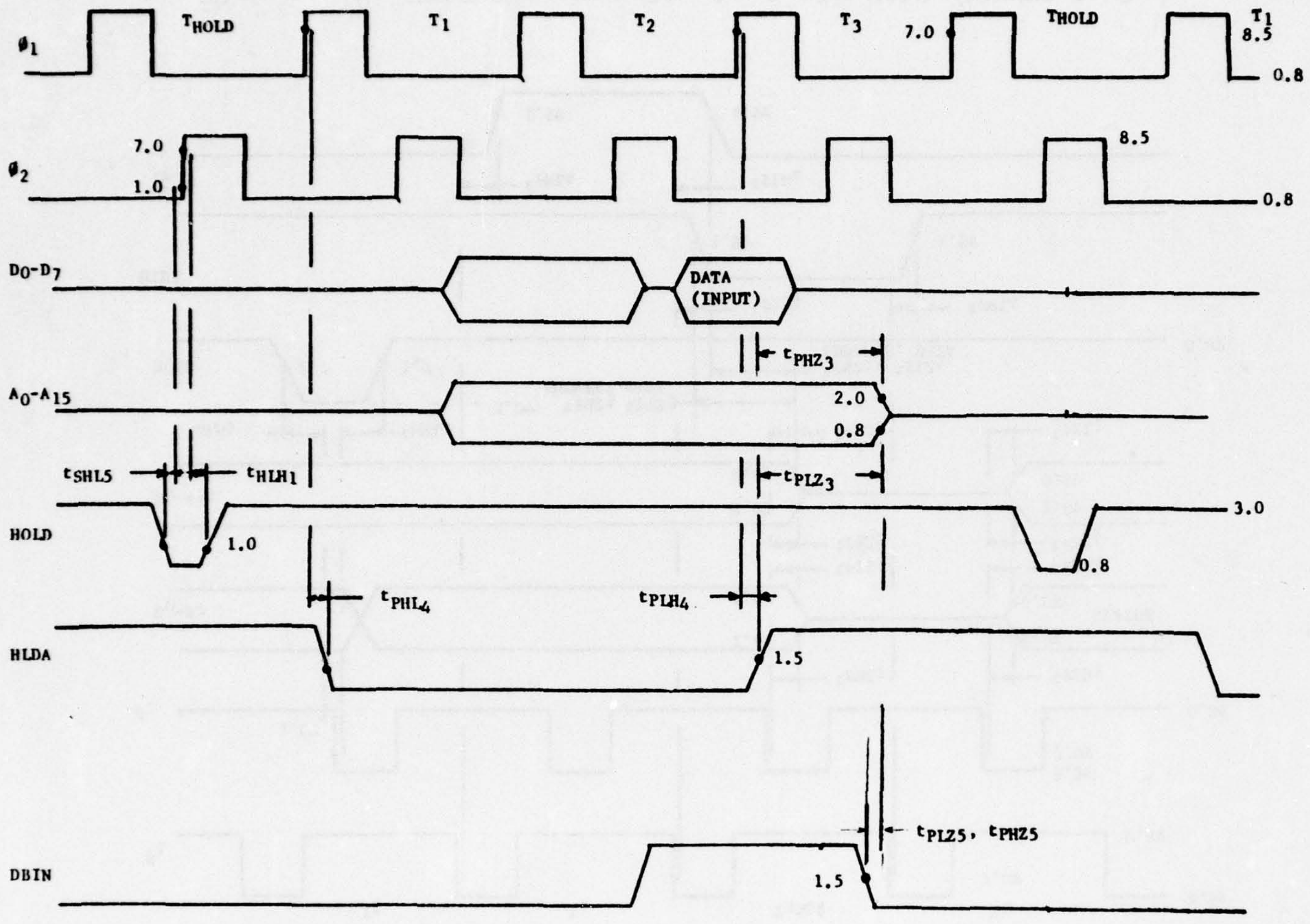


Figure 4-7. HOLD Sequence-Single Machine Cycle EXECUTE (READ Mode)

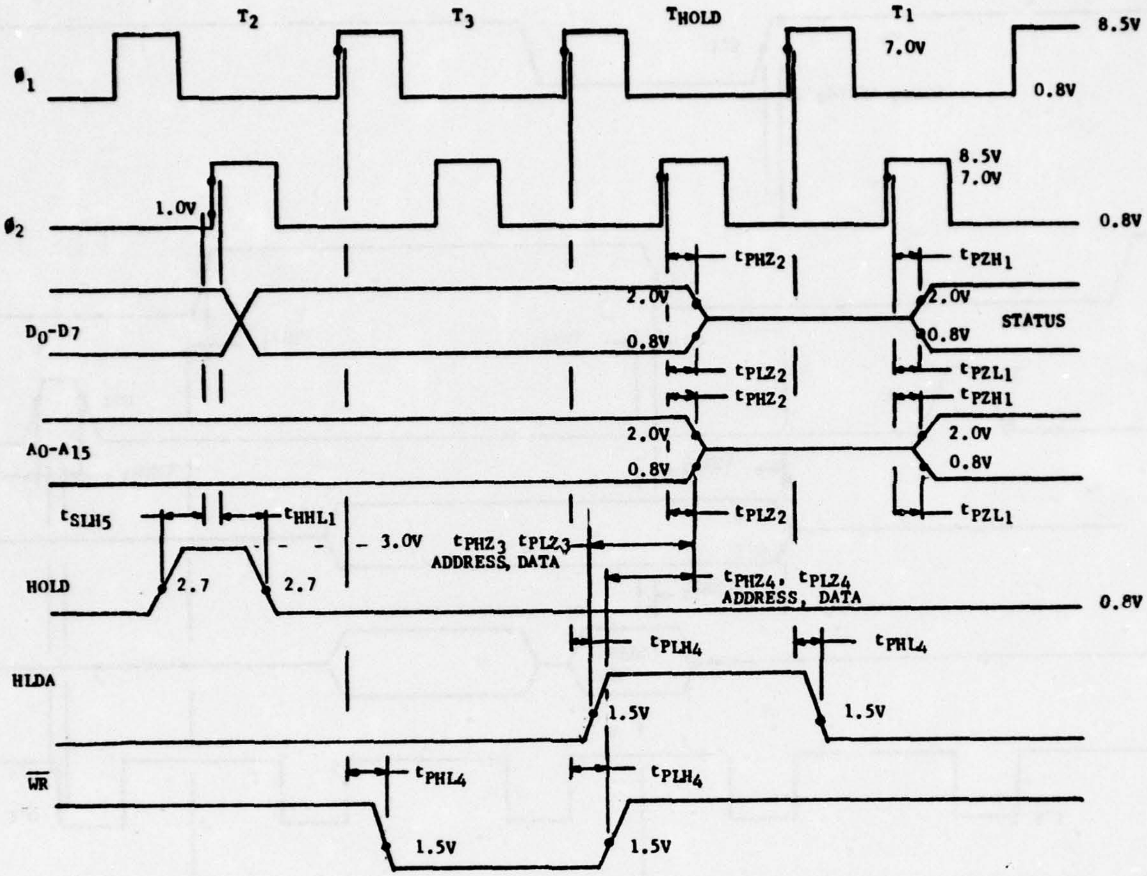


Figure 4-8. HOLD Sequence-Single Cycle HOLD (WRITE Cycle)

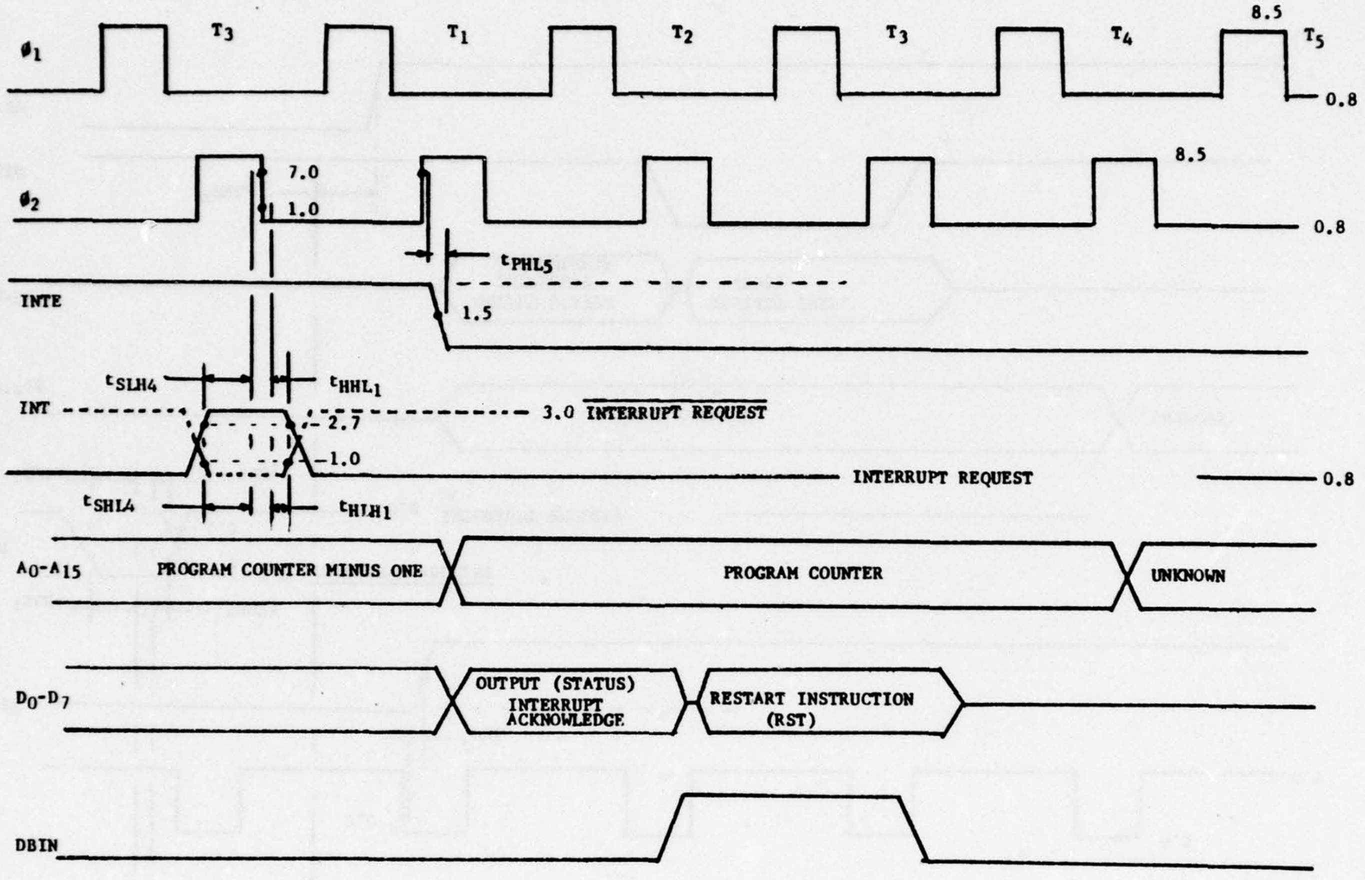


Figure 4-9. INTERRUPT Timing

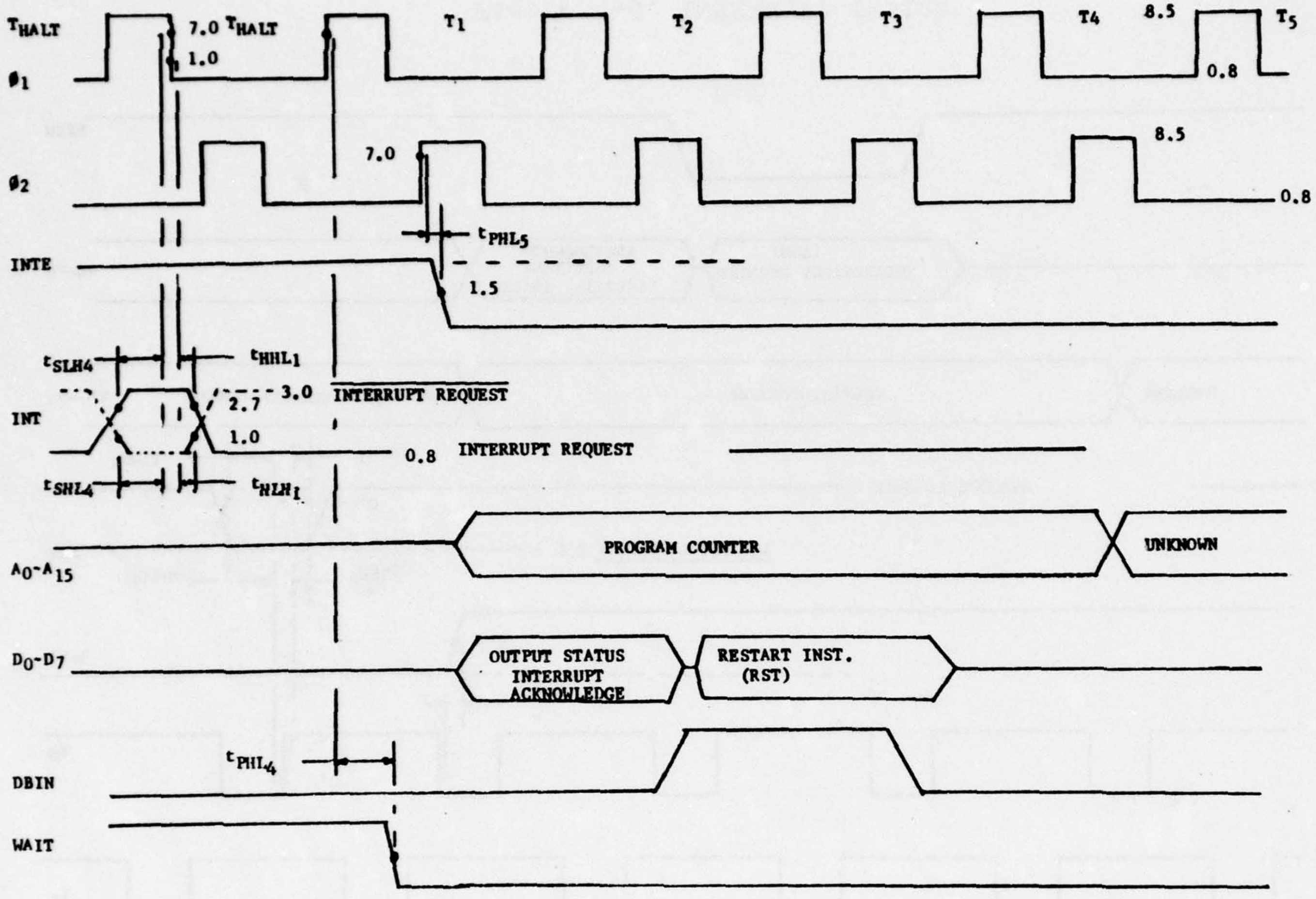


Figure 4-10. INTERRUPT Timing in HALT Mode

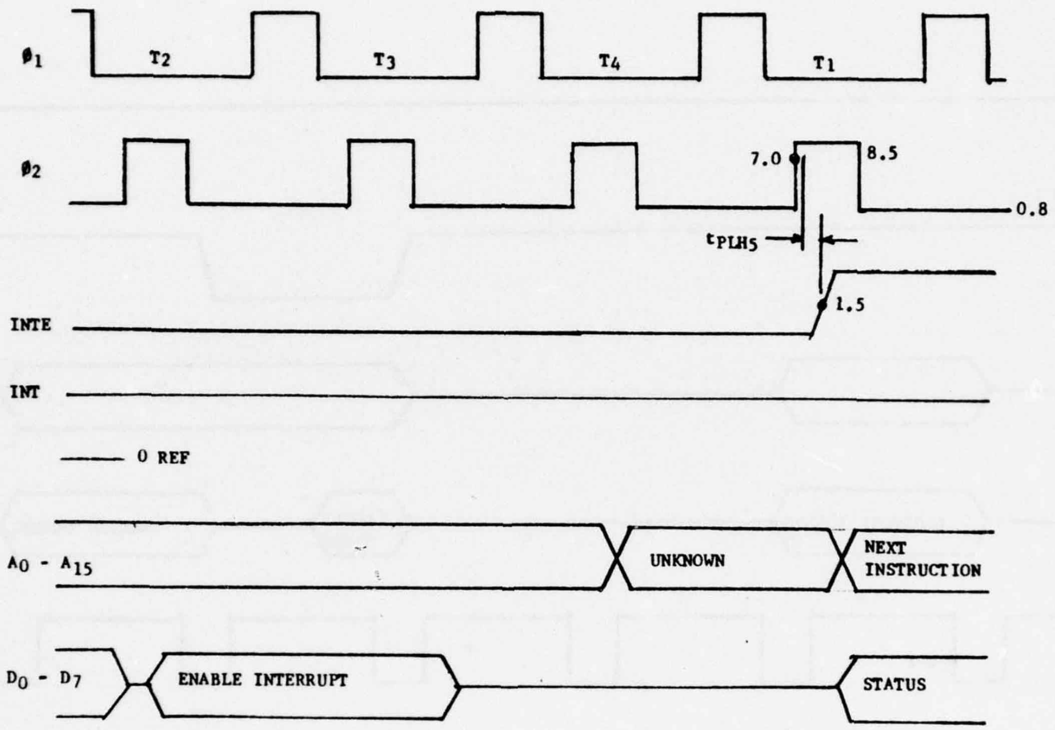


Figure 4-11. INTERRUPT ENABLE Output-Set INTE Flip-flop

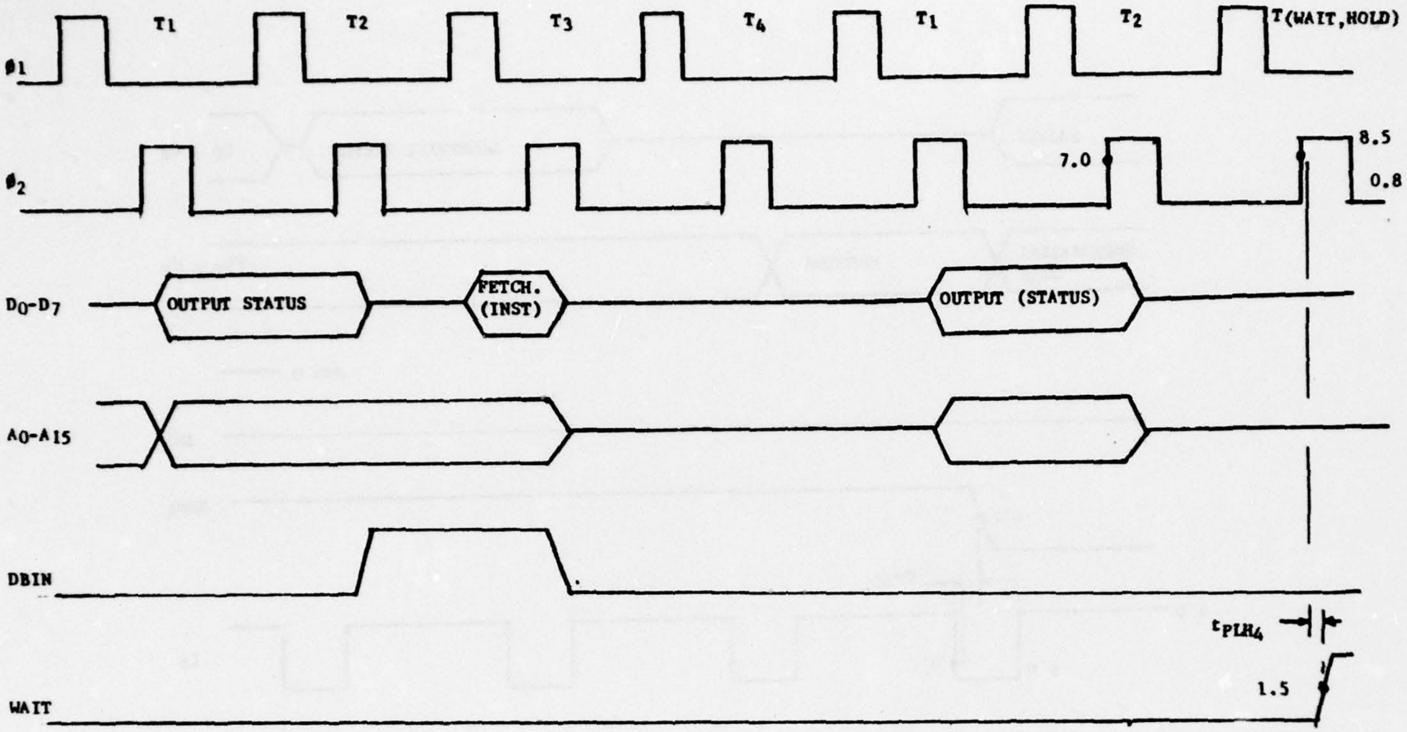


Figure 4-12. HALT Timing

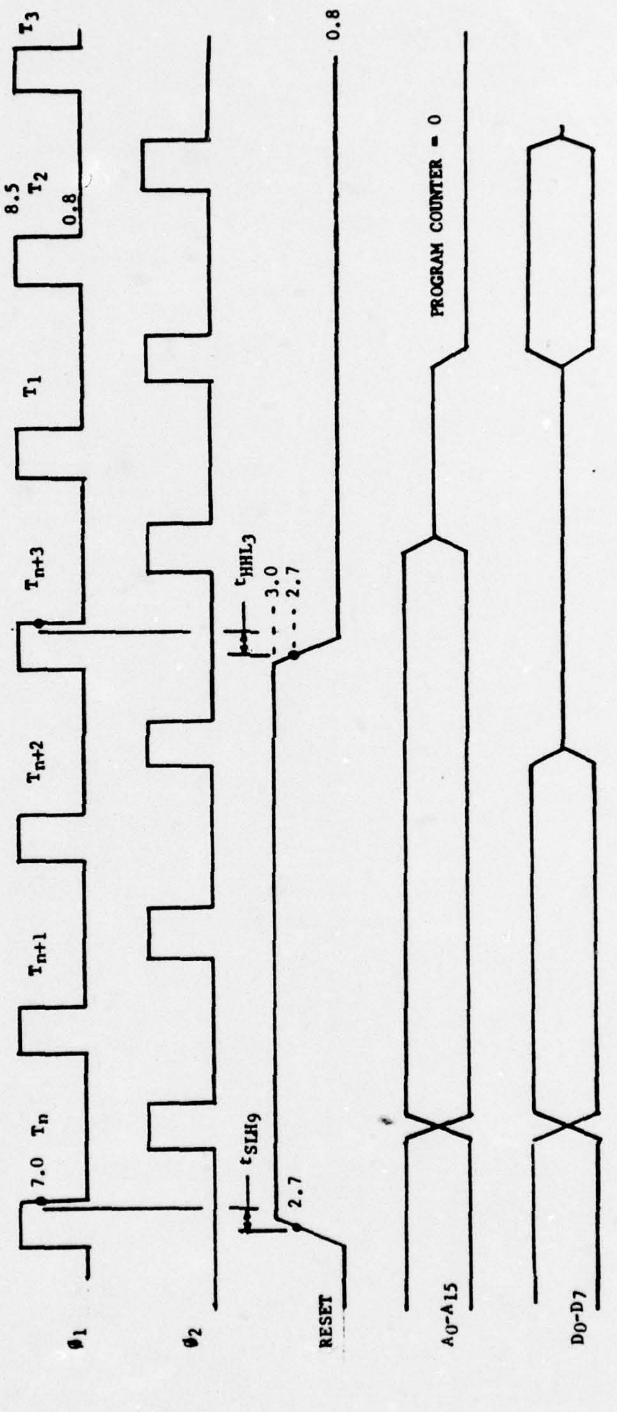


Figure 4-13. RESET Timing

SECTION V

TABLE OF CONTENTS

SECTION V

SUMMARY OF LOGIC INTEGRITY TESTS FOR MSI DEVICES

The Logic Integrity Test (LIT) is a functional test generated for Medium Scale Integration (MSI) and Large Scale Integration (LSI) devices that checks each internal gate node for a stuck at logic "1" and stuck at logic "0" condition. The "stuck at" conditions of the internal gates can be checked by exercising the device input terminals with test patterns of ones and zeros (test vectors).

This volume consists of Logic Integrity Test (LIT) reports for tests generated for the integrated circuit devices listed in the index. All of the tests are for devices that either are already included in MIL-M-38510 slash sheets or are to be included in future slash sheets. LIT's were generated for nine low-power Schottky TTL flip-flops and one CMOS shift register. For the CMOS device, additional tests were generated to check for worst case leakage paths.

In most instances the logic diagrams of the low-power Schottky devices were identical to other TTL flip-flops which had been analyzed in a previous contract with RADC*. Although logic diagrams were the same, the circuit schematics did vary and in some cases the functions assigned to the integrated circuit package pins varied as well.

The test vector sets that were generated provide for a testing confidence level (TCL) of 100 percent. That is, all the tests that are required to completely test the device are accomplished. All the tests were verified by testing devices to the recommended vectors.

*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

REPORT NO. 5.1

Logic Integrity Test for the Worst Case Leakage Test for Vendor D CD4035A COS/MOS Four-stage Parallel In/Parallel Out Shift Register

OBJECTIVE:

Develop a Logic Integrity Test (LIT) and Worst Case Leakage Test for the CD 4035A, COS/MOS 4-Stage Parallel In/Parallel Out Shift Register.

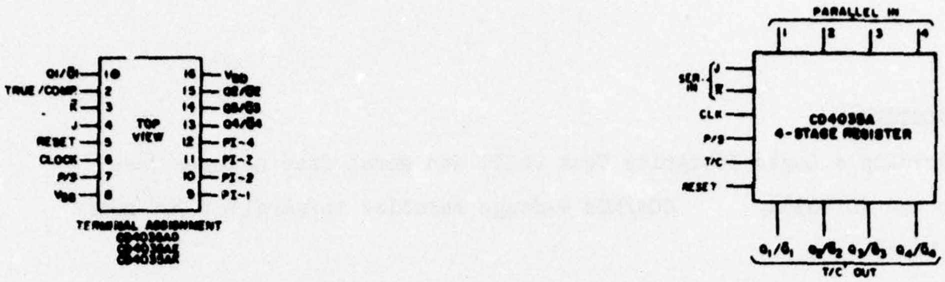
CIRCUIT DESCRIPTION:

The CD 4035A (Figure 1) is a four-stage clocked serial register having provisions for synchronous parallel inputs to each stage and serial inputs to the first stage via $J\overline{K}$ logic. Register stages 2, 3 and 4 are coupled in a serial "D" flip-flop configuration when the register is in the serial mode (Parallel/Serial control low).

Parallel entry via the "D" line of each register stage is permitted only when the Parallel/Serial control is "high". In the parallel or serial mode, information is transferred on positive clock transitions.

When the True/Complement control is "high", the True contents of the register are available at the output terminals. When the True/Complement control is "low", the outputs are the complements of the data in the register. The True/Complement control functions asynchronously with respect to the clock signal.

$J\overline{K}$ input logic is provided on the first stage serial input to minimize logic requirements particularly in counting and sequence generation applications. With $J\overline{K}$ inputs connected together, the first stage becomes a "D" flip-flop. An asynchronous common reset is also provided.



(A)
BLOCK DIAGRAM

(B)
FUNCTIONAL DIAGRAM

Fig. 1

TEST PLAN:

1. Review the manufacturer's specification and verify the Logic Flow Diagram with its Schematic Diagram.
2. Generate a Test Vector Set.
3. Determine a Testing Confidence Level (TCL).
4. Test device with generated test vectors.
5. Develop Leakage Test.

SUMMARY:

The CD 4035A specification was reviewed and the Logic Flow Diagram was found to be an accurate description of the Schematic Diagram. A test set was generated and a TCL of 100% was attained. Four devices were successfully tested for both leakage and LIT. All four devices passed the LIT test, while two of the four failed the suggested military specification for leakage.

ANALYSIS:

A. Logic Integrity Test (LIT)

A single stage of the CD 4035A was evaluated based on a Stuck-at-one (S-A-1), Stuck-at-zero (S-A-0) test philosophy.

The Logic Diagram, Figure 2, depicts a "D"-type flip-flop with reset capability.

The test requirements for the "D"-type flip-flop were discussed in the report on the CD 4015A, COS/MOS Dual 4-Stage Shift Register.

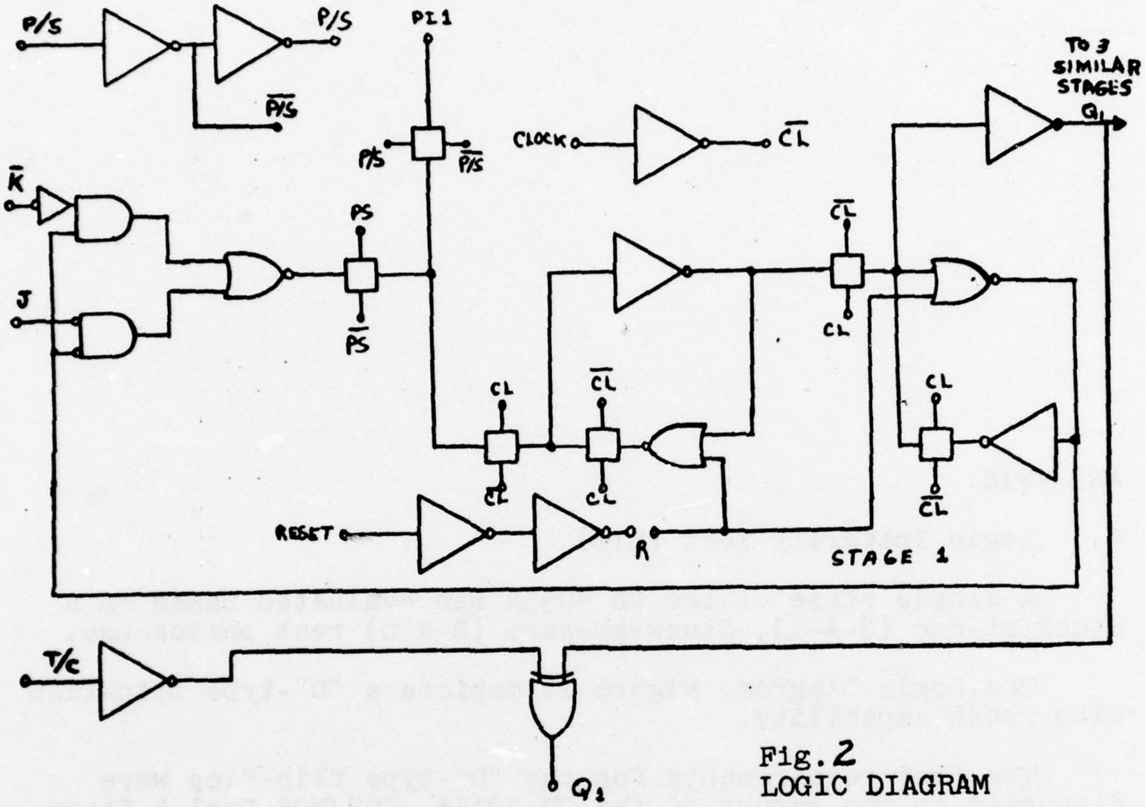


Fig. 2
LOGIC DIAGRAM

The test requirements for the two input EXCLUSIVE OR gate are presented in Table 1.

A	B	C	A	B
0	0	0	S-A-1	S-A-1
0	1	1	S-A-1	S-A-0
1	0	1	S-A-0	S-A-1
1	1	0	S-A-0	S-A-0

Table 1

Table 1 satisfies the requirements for a S-A-0 and S-A-1 for both inputs if either vectors 2 and 3 or 1 and 4 are used. However the Exclusive OR gate is a combination of "inverters" and transmission gates. This combination requires that all vectors be used.

A set of test vectors (Table 2) was developed for a single stage of the Shift Register.

The test set for the complete device was then developed. This effort consisted of verifying that a failure will be propagated to an output. The test vectors required to test the CD 4035A are shown in Table 3.

Test Cond.	Parallel Input	P/S	J	$\overline{K}$	Clock	Reset	T/C	Q
1	1	1	0	0	1	1	1	0
2	1	1	0	0	1	0	1	0
3	1	1	0	0	0	0	1	0
4	1	1	0	0	1	0	1	1
5	1	0	0	0	1	0	1	1
6	1	0	0	0	0	0	1	1
7	1	0	0	0	1	0	1	0
8	0	0	1	0	1	0	1	0
9	0	0	1	0	0	0	1	0
10	0	0	1	0	1	0	1	1
11	0	0	0	1	0	0	1	1
12	0	0	0	1	1	0	1	1
13	0	1	0	1	1	0	1	1
14	0	1	0	1	0	0	1	1
15	0	1	0	1	1	0	1	0
16	1	0	0	0	0	0	0	1
17	1	0	0	0	1	0	0	1
18	1	1	0	0	0	0	0	1
19	1	1	0	0	1	0	0	0
20	1	1	0	0	0	0	0	0
21	1	1	0	0	0	1	0	1
22	1	1	0	0	1	1	0	1

Table 2

Test Vectors For Single Stage

TABLE 3

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	Q_1/Q_1	T/C	$\overline{K}$	J	RE-SET	CL	P/S	V_{SS}	PI-1	PI-2	PI-3	PI-4	Q_4/Q_4	Q_3/Q_3	Q_2/Q_2	V_{DD}		Min	Max	Min	Max	Min	Max	
7		3014	01	X	5V	GND	GND	GND	GND	5V	GND	5V	5V	5V	5V	X	X	X	5V	All Outputs	H or L As shown		-	-	-	-	Vdc
			02	H	5V	GND	GND	GND	5V	5V		5V	5V	5V	5V	H	H	H					-	-	-	-	
			03	H	5V	GND	GND	GND	GND	5V		5V	5V	5V	5V	H	H	H					-	-	-	-	
			04	L	5V	GND	GND	5V	GND	5V		5V	5V	5V	5V	L	L	L					-	-	-	-	
			05	L	5V	GND	GND	5V	5V	5V		5V	5V	5V	5V	L	L	L					-	-	-	-	
			06	L	5V	GND	GND	GND	5V	5V		5V	GND	5V	GND	L	L	L					-	-	-	-	
			07	L	5V	GND	GND	GND	GND	5V		5V	GND	5V	GND	L	L	L					-	-	-	-	
			08	H	5V	GND	GND	GND	5V	5V		5V	GND	5V	GND	L	H	L					-	-	-	-	
			09	H	5V	GND	GND	GND	5V	GND		5V	GND	5V	GND	L	H	L					-	-	-	-	
			10	H	5V	GND	GND	GND	GND	GND		5V	GND	5V	GND	L	H	L					-	-	-	-	
			11	L	5V	GND	GND	GND	5V	GND		5V	GND	5V	GND	H	L	H					-	-	-	-	
			12	L	5V	GND	5V	GND	5V	GND		GND	5V	GND	5V	H	L	H					-	-	-	-	
			13	L	5V	GND	5V	GND	GND	GND		GND	5V	GND	5V	H	L	H					-	-	-	-	
			14	H	5V	GND	5V	GND	5V	GND		GND	5V	GND	5V	L	H	L					-	-	-	-	
			15	H	5V	5V	GND	GND	GND	GND		GND	5V	GND	5V	L	H	L					-	-	-	-	
			16	H	5V	5V	GND	GND	5V	GND		GND	5V	GND	5V	H	L	H					-	-	-	-	
			17	L	5V	GND	5V	5V	GND	5V		GND	5V	GND	5V	L	L	L					-	-	-	-	
			18	L	5V	GND	5V	GND	GND	5V		GND	5V	GND	5V	L	L	L					-	-	-	-	
			19	L	5V	GND	5V	GND	5V	5V		GND	5V	GND	5V	H	L	H					-	-	-	-	
7		3014	20	L	5V	GND	GND	5V	GND	5V	GND	5V	5V	5V	5V	L	L	L	5V	All Outputs	H or L As shown		-	-	-	-	Vdc

TABLE 3 (concluded)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	$Q1/\overline{Q1}$	T/C	$\overline{K}$	J	RE-SET	CL	P/S	V_{SS}	PI-1	PI2	PI-3	PI-4	$Q4/\overline{Q4}$	$Q3/\overline{Q3}$	$Q2/\overline{Q2}$	V_{DD}		Min	Max	Min	Max	Min	Max	
7		3014	21	H	GND	GND	GND	GND	GND	5V	GND	5V	5V	5V	5V	H	H	H	5V	All Outputs	H or	L	-	-	-	-	Vdc
↓		↓	22	L	GND	GND	GND	GND	5V	5V	↓	5V	5V	5V	5V	L	L	L	↓	↓	As shown		-	-	-	-	↓
↓		↓	23	L	GND	5V	5V	GND	GND	5V	↓	GND	GND	GND	GND	L	L	L	↓	↓	↓	↓	-	-	-	-	↓
↓		↓	24	H	GND	5V	5V	GND	5V	5V	↓	GND	GND	GND	GND	H	H	H	↓	↓	↓	↓	-	-	-	-	↓
↓		↓	25	H	GND	GND	GND	5V	GND	GND	↓	5V	5V	5V	5V	H	H	H	↓	↓	↓	↓	-	-	-	-	↓
↓		↓	26	H	GND	GND	GND	GND	GND	GND	↓	5V	5V	5V	5V	H	H	H	↓	↓	H or	L	-	-	-	-	↓
7		3014	27	H	GND	GND	GND	GND	5V	GND	GND	5V	5V	5V	5V	H	H	H	5V	All Outputs	As shown		-	-	-	-	Vdc

NOTES:

1. L = 0.01V Max., H = 4.99V Min.
2. Test must run in sequence shown.
3. X = "don't care" condition.

B. Leakage Test

Using the single analysis, a leakage pattern was developed for the complete device. Table 4 is one possible sequence to obtain the test for leakage.

C. Test Mechanization for Leakage

When performing the leakage test the Logic "1" and "0" input voltage must be V_{DD} and V_{SS} respectively. This is required to detect the contribution of the Input Transmission Gate Leakages. The test circuit is shown in Figure 3.

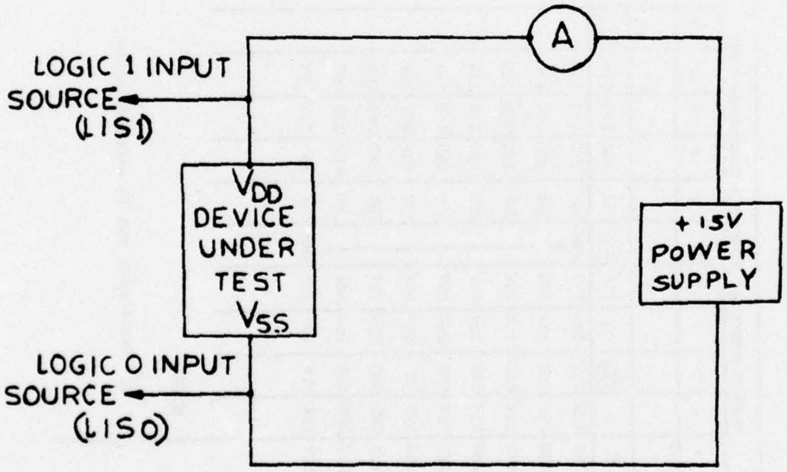


Figure 3

LEAKAGE CURRENT TEST

An, input is connected to LIS1 if a Logic "1" is specified or to LISO is a Logic "0" is specified.

TABLE 4

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		Subgroup 1 $T_A = 25°C$		Subgroup 2 $T_A = 125°C$		Subgroup 3 $T_A = -55°C$		
			TEST NO.	$Q_1/\overline{Q}_1$	$T/\overline{C}$	$\overline{K}$	J	RE-SET	CL	P/S	V_{SS}	PI-1	PI-2	PI-3	PI-4	$Q_4/\overline{Q}_4$	$Q_3/\overline{Q}_3$	$Q_2/\overline{Q}_2$	V_{DD}		Min	Max	Min	Max	Min	Max	
	I_{SS}	3005	001		15V	15V	GND	15V	GND	15V	GND	15V	15V	15V	15V				15V	V_{DD}		500		5000			nAdc
	↓	↓	002		15V	GND	15V	15V	15V	15V	↓	GND	GND	GND	GND				↓	NONE		↓		↓			↓
	↓	↓	003		15V	GND	15V	GND	15V	15V	↓	GND	GND	GND	GND				↓	V_{DD}		↓		↓			↓
	↓	↓	004		15V	GND	15V	GND	15V	15V	↓	15V	15V	15V	15V				↓	NONE		↓		↓			↓
	↓	↓	005		15V	GND	15V	GND	GND	15V	↓	15V	15V	15V	15V				↓	NONE		↓		↓			↓
	↓	↓	006		15V	GND	15V	GND	15V	15V	↓	15V	15V	15V	15V				↓	V_{DD}		↓		↓			↓
	↓	↓	007		15V	15V	GND	GND	15V	15V	↓	GND	GND	GND	GND				↓	NONE		↓		↓			↓
	↓	↓	008		GND	15V	GND	GND	15V	GND	↓	GND	GND	GND	GND				↓	V_{DD}		↓		↓			↓
	I_{SS}	3005	009		GND	15V	15V	15V	15V	15V	GND	15V	15V	15V	15V				15V	V_{DD}		500		5000			nAdc

NOTES:

1. Tests must run in sequence show

D. Leakage Test Results

Tne test setup of Figure 5 was used to accomplish the leakage test on the CD 4035A. When performing the test, the clock input was driven from an external flip-flop where the power supply was separate from V_{DD}. Tne clock input represents a gate leakage and, therefore, is not required to be connected to LIS1 or LISO.

INPUTS										OUTPUTS				DEVICES				
P/S	PI-1	PI-2	PI-3	PI-4	J	$\overline{K}$	T/C	RE-SET	CL	$Q_1/\overline{Q}_1$	$Q_2/\overline{Q}_2$	$Q_3/\overline{Q}_3$	$Q_4/\overline{Q}_4$	#1	#2	#3	#4	UNITS
1	1	1	1	1	0	1	1	1	0	0	0	0	0	3000	0.5	0.4	3.5	NANO AMPS
1	0	0	0	0	1	0	1	0	1	0	0	0	0	2900	0.5	0.3	2.0	↓
1	1	1	1	1	1	0	1	0	1	1	1	1	1	170	0.7	0.4	3.8	↓
0	0	0	0	0	0	1	0	0	1	0	0	0	0	3000	0.75	510	3.2	↓
1	1	1	1	1	1	1	0	1	1	1	1	1	1	150	0.7	0.4	3.9	NANO AMPS

Device #1 and #3 exhibited high leakage, greater than 500 nanoamps. These two devices passed the LIT Test.

REPORT NO. 5.2

Logic Integrity Test for the 54LS73 (Vendor C) Dual J-K Negative Edge-Triggered Flip-Flop

Objective:

Develop a Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 01 (54LS73 Dual J-K Edge Triggered Flip Flop).

Circuit Description

The 54LS73 is a dual J-K Edge-Triggered Flip-Flop with Asynchronous Clear. When the clock goes high, the inputs are enabled and the data will be accepted. The logic level of the J and K inputs may be allowed to change while the clock pulse is high as long as minimum set up time is observed. Input data is transferred to the outputs on the negative going edge of the clock pulse.

Test Plan

I. Review the Logic Flow Diagram presented in the manufacturer's specification (Figure 1B), and the Schematic Diagram in MIL-M-38510/301 (Figure 1C) to determine if:

a. The circuit mechanization presented is functionally correct, and

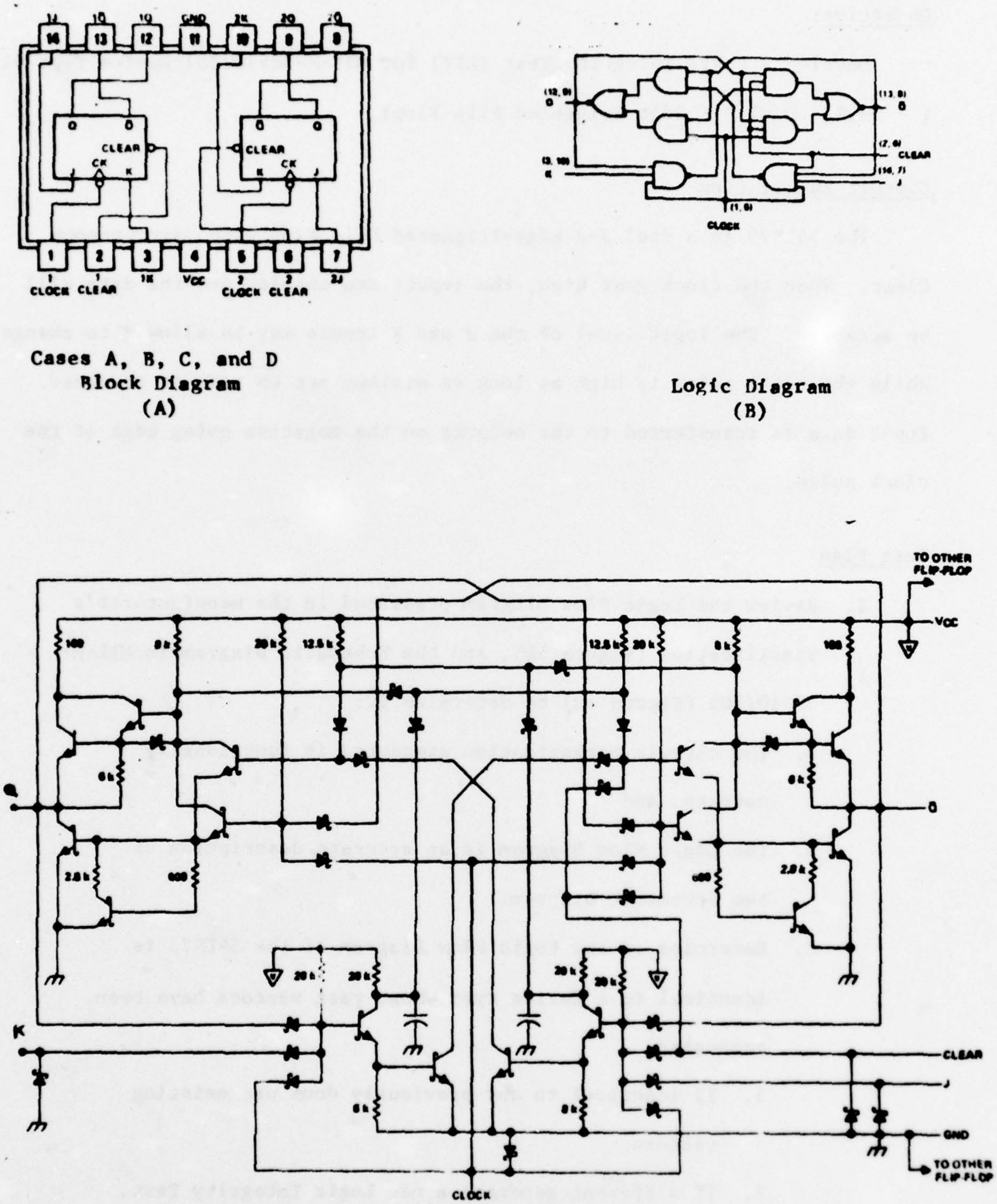
b. The Logic Flow Diagram is an accurate description of the Schematic Diagram.

c. Determine if the Logic Flow Diagram of the 54LS73 is identical to a device type whose test vectors have been generated.

1. If identical to any previously done use existing vectors.

2. If different generate a new Logic Integrity Test.

II. Test a device with the vectors.



Cases A, B, C, and D
Block Diagram
(A)

Logic Diagram
(B)

SCHEMATIC DIAGRAM
(C)

FIGURE 1

Summary:

The Schematic Diagram and Logic Flow Diagram for the 54LS73 were reviewed. The circuit mechanization was found to be functionally correct. The Logic Flow Diagram requires the addition of two delays to operate properly and represent the true mechanization of the Schematic Diagram. The reasons for the addition of the delays is discussed in the Analysis section.

It was determined that the 54LS73 has a fundamentally similar transistor diagram to the 54S113. If the pin names of the 54S113 were changed as in Table 1, then the 54S113 is logically identical to the 54LS73.

PIN NAMES OF 54S113	PIN NAMES OF 54LS73
Q	$\overline{Q}$
$\overline{Q}$	Q
J	K
K	J
PRESET	CLEAR

TABLE 1

Therefore, it is recommended that the 54LS73 be tested using the vectors developed for the 54S113* (Table 2). These vectors provide a 100% TCL for both devices.

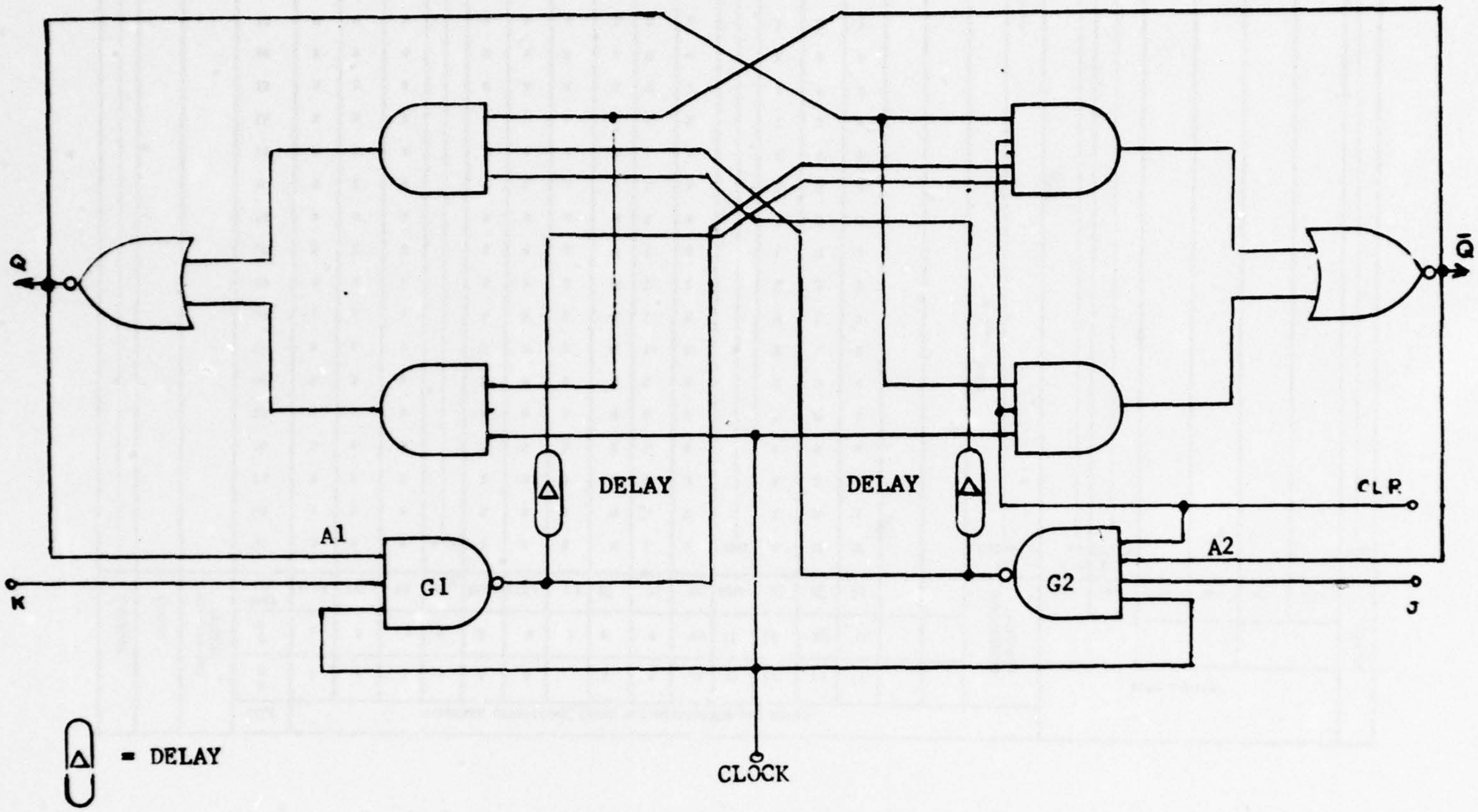
A 54LS73 was successfully tested using these vectors.

*RADC-TR-75-216, Volume II, August 1975, (A017314). Digital Microcircuit Characterization and Specification

Analysis

Analysis of the circuit operation of the 54LS73 indicated that proper functioning requires relatively long propagation delays in gates G1 and G2 (Figure 2). These delays are necessary to allow input information from gates G1 and G2 to be accepted by the latch on a negative transition of the clock before the low level of the clock disables G1 and G2. Absence of these delays may result in an indeterminate state. These delays are incorporated in the Schematic Diagram. It is recommended that these delays be reflected in the Logic Diagram as presented in Figure 2.

There are two possible faults in each flip-flop which lead to an indeterminate state. These are S-A-1 faults on inputs A1 and A2 to gates G1 and G2 respectively (Figure 2). If either of these faults exists and the timing is such that the function is affected, the fault will be caught by the vectors presented in Table 2. If the internal gate delays are such that the fault does not affect the function, the fault is considered irrelevant. With this qualification, a TCL of 100% is achieved with the vectors presented in Table 2.



RECOMMENDED LOGIC DIAGRAM

FIGURE 2

TABLE 2

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	1CLK	1CLR	1K	V_{CC}	2CLK	2CLR	2J	$2\bar{Q}$	2Q	2K	GND	1Q	$1\bar{Q}$	1J				Min	Max	Min	Max	Min	Max	
7			01	B	B	B	4.5V	B	B	B	H	L	B	GND	L	H	A			ALL Outputs	H or L as shown						VDC
			02	A	B	B		B	B	B	H	L	B		L	H	A										
			03	B	B	B		B	B	B	H	L	B		L	H	A										
			04	B	A	B		B	B	B	H	L	B		L	H	A										
			05	A	A	B		B	B	B	H	L	B		L	H	A										
			06	B	A	B		B	B	B	H	L	B		H	L	A										
			07	B	A	A		B	B	B	H	L	B		H	L	B										
			08	A	A	A		B	B	B	H	L	B		H	L	B										
			09	B	A	A		B	B	B	H	L	B		L	H	B										
			10	B	B	B		B	B	A	H	L	B		L	H	B										
			11	B	B	B		B	B	A	H	L	B		L	H	B										
			12	B	B	B		B	B	A	H	L	B		L	H	B										
			13	B	B	B		B	A	A	H	L	B		L	H	B										
			14	B	B	B		A	A	A	H	L	B		L	H	B										
			15	B	B	B		B	A	A	L	H	B		L	H	B										
			16	B	B	B		B	A	B	L	H	A		L	H	B										
			17	B	B	B		A	A	B	L	H	A		L	H	B										
			18	B	B	B		B	A	B	H	L	A		L	H	B										
			19	B	B	B		B	B	B	H	L	B		L	H	B										
7			20	B	A	B	4.5V	B	A	B	H	L	B	GND	L	H	B			ALL Outputs	H or L as shown						VDC

TABLE 2 (Cont'd)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	1CLK	1CLR	1K	V_{CC}	2CLK	2CLR	2J	$2\bar{Q}$	2Q	2K	GND	1Q	$1\bar{Q}$	1J				Min	Max	Min	Max	Min	Max	
7			21	A	A	B	4.5V	A	A	B	H	L	B	GND	L	H	B			ALL Outputs	H or L as shown						VDC
			22	B	A	B		B	A	B	H	L	B		L	H	B										
			23	B	A	A		B	A	A	H	L	A		L	H	A										
			24	A	A	A		A	A	A	H	L	A		L	H	A										
			25	B	A	A		B	A	A	L	H	A		H	L	A										
			26	B	A	B		B	A	B	L	H	B		H	L	B										
			27	A	A	B		A	A	B	L	H	B		H	L	B										
			28	B	A	B		B	A	B	L	H	B		H	L	B										
			29	B	A	A		B	A	A	L	H	A		H	L	A										
			30	A	A	A		A	A	A	L	H	A		H	L	A										
7			31	B	A	A	4.5V	B	A	A	H	L	A	GND	L	H	A			ALL Outputs	H or L as shown						VDC

NOTES:

1. L = 0.4V max.
   H = 2.4V min.

2. A = 2.0V min.
   B = 0.7V max.

3. Test must be run in sequence shown.

REPORT NO. 5.3

Logic Integrity Test for the 54LS74 Dual D-type Positive Edge-Triggered Flip-Flop

Objective:

Develop a Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 02 (54LS74 Dual D-Type Positive-Edge-Triggered Flip-Flop with Preset and Clear).

Circuit Description:

The 54LS74 (Figure 1) is a monolithic, dual, D-type, edge-triggered flip-flop using Schottky TTL circuitry. It features direct clear and preset inputs and complementary Q and $\overline{Q}$ outputs. Input information is transferred to the outputs on the positive edge of the clock pulse.

Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive going pulse. After the clock input threshold voltage has been passed, the data input (D) is locked out.

Test Plan:

I. Review the Logic Flow Diagram presented in the manufacturer's specification (Figure 1B), and the Schematic Diagram in MIL-M-38510/301 (Figure 1C) to determine if:

   a. The circuit mechanization presented is functionally correct, and

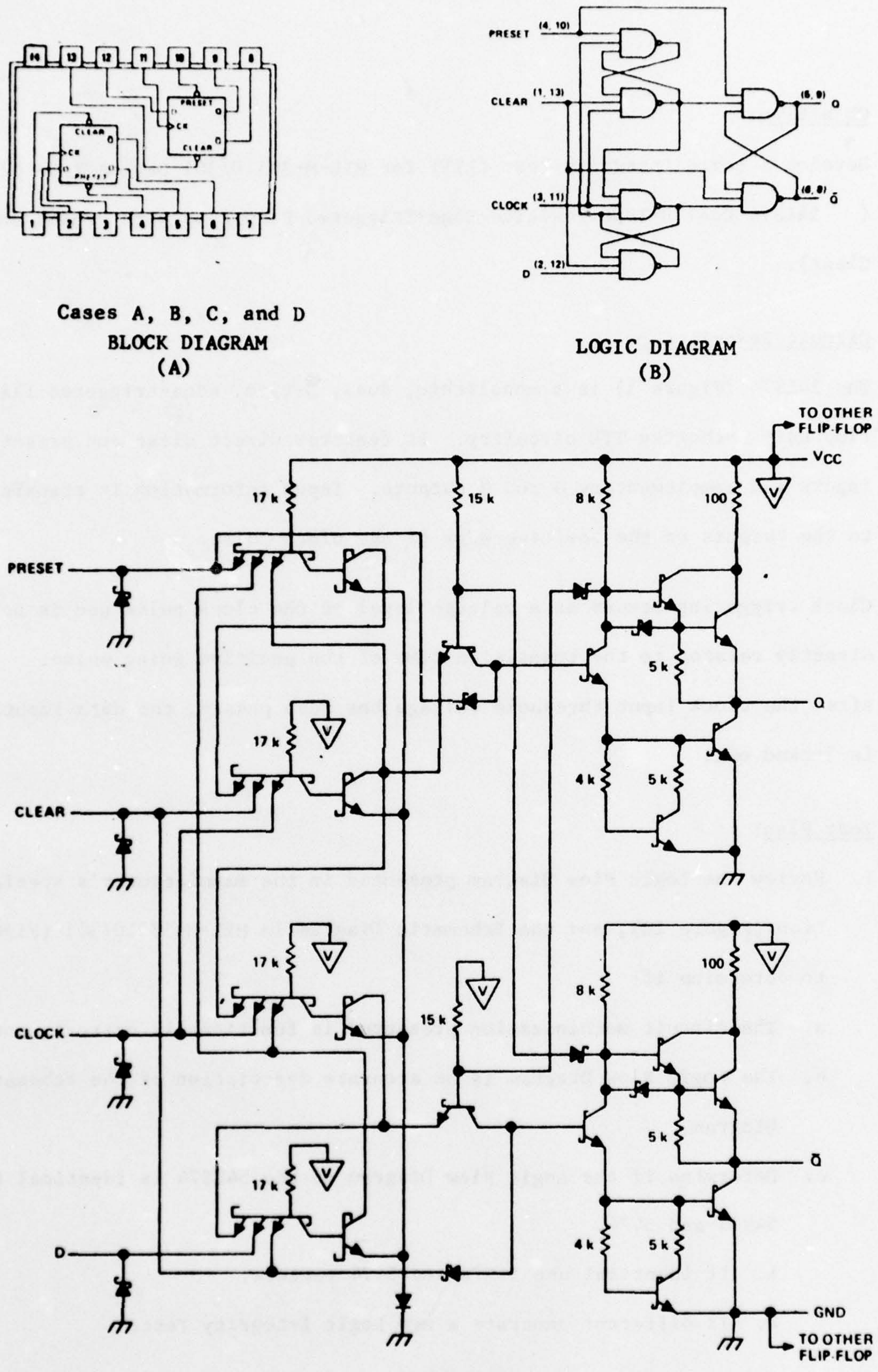
   b. The Logic Flow Diagram is an accurate description of the Schematic Diagram.

   c. Determine if the Logic Flow Diagram of the 54LS74 is identical to the 54S74 and 5474.

      1. If identical use 54S74 and 5474 vectors.

      2. If different generate a new Logic Integrity Test.

II. Test a device with the vectors.



Cases A, B, C, and D
BLOCK DIAGRAM
(A)

LOGIC DIAGRAM
(B)

SCHEMATIC DIAGRAM
(C)

FIGURE 1

Summary

The Schematic Diagram and Logic Flow Diagram for the 54LS74 were reviewed. The circuit mechanization was functionally correct.

It was determined that the 54LS74 is logically identical to the 5474 and 54S74, and has a fundamentally similar transistor diagram. It is recommended that it be tested using vectors developed for the 54S74 and 5474* (Table 1). These vectors provide a 100% TCL for all three devices.

A 54LS74 was successfully tested using these vectors.

*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

TABLE 1

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	MEASUREMENT TERMINAL							
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	1CLR	1D	1CK	1PR	1Q	$\overline{1Q}$	GND	$\overline{2Q}$	2Q	2PR	2CK	2D	2CLR	VCC				Min	Max	Min	Max	Min	Max	
7			1	B	B	B	B	H	H	GND	H	H	B	B	B	B	4.5V			ALL Outputs	H or L as shown						VDC
			2	B	B	B	A	L	H		H	L	A	B	B	B											
			3	A	B	B	A	L	H		H	L	A	B	B	A											
			4	A	B	B	B	H	L		L	H	B	B	B	A											
			5	A	B	A	B	H	L		L	H	B	A	B	A											
			6	B	B	A	B	H	H		H	H	B	A	B	B											
			7	B	A	A	B	H	H		H	H	B	A	A	B											
			8	B	A	A	A	L	H		H	L	A	A	A	B											
			9	A	A	A	A	L	H		H	L	A	A	A	A											
			10	A	A	A	B	H	L		L	H	B	A	A	A											
			11	A	A	A	A	H	L		L	H	A	A	A	A											
			12	A	A	B	A	H	L		L	H	A	B	A	A											
			13	A	B	B	A	H	L		L	H	A	B	B	A											
			14	A	B	A	A	L	H		H	L	A	A	B	A											
			15	A	B	A	B	H	L		L	H	B	A	B	A											
			16	B	A	A	B	H	H		H	H	B	A	A	B											
			17	B	B	A	B	H	H		H	H	B	A	B	B											
			18	B	B	A	A	L	H		H	L	A	A	B	B											
			19	A	B	A	A	L	H		H	L	A	A	B	A											
7			20	A	A	B	A	L	H	GND	H	L	A	B	A	A	4.5V			ALL Outputs	H or L as shown						VDC

TABLE 1 (Cont'd.)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	1CLR	1D	1CK	1PR	1Q	$1\bar{Q}$	GND	$2\bar{Q}$	2Q	2PR	2CK	2D	2CLR	VCC				Min	Max	Min	Max	Min	Max	
7			21	A	A	A	A	H	L	GND	L	H	A	A	A	A	4.5V			ALL Outputs	H or L as shown						VDC
↓			22	A	A	A	B	H	L	↓	L	H	B	A	A	A	↓			↓	↓	↓					↓
↓			23	A	A	A	A	H	L	↓	L	H	A	A	A	A	↓			↓	↓	↓					↓
↓			24	B	A	A	A	L	H	↓	H	L	A	A	A	B	↓			↓	↓	↓					↓
↓			25	A	A	A	A	L	H	↓	H	L	A	A	A	A	↓			↓	↓	↓					↓
↓			26	A	B	A	B	H	L	↓	L	H	B	A	B	A	↓			↓	↓	↓					↓
7			27	A	B	A	A	H	L	GND	L	H	A	A	B	A	4.5V			ALL Outputs	H or L as shown						VDC

NOTES:

1. L = 0.4V max. H = 2.4V min.
2. A = 2.0V min. B = 0.7V max.
3. Test must be run in sequence shown.

REPORT NO. 5.4

Logic Integrity Test for the 54LS107 (Vendor C) Dual J-K Negative Edge-Triggered Flip-Flop

Objective:

Develop a Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 08 (54LS107 Dual J-K Negative Edge Triggered Flip Flop with Clear).

Circuit Description

The SN54LS107 is a dual J-K Edge-Triggered Flip-Flop with Asynchronous Clear. When the clock goes high, the inputs are enabled and the data will be accepted. The logic level of the J and K inputs may be allowed to change while the clock pulse is high as long as minimum set up time is observed. Input data is transferred to the outputs on the negative going edge of the clock pulse.

Test Plan

I. Review the Logic Flow Diagram presented in the manufacturer's specification (Figure 1B), and the Schematic Diagram in MIL-M-38510/301 (Figure 1C) to determine if:

   a. The circuit mechanization presented is functionally correct, and

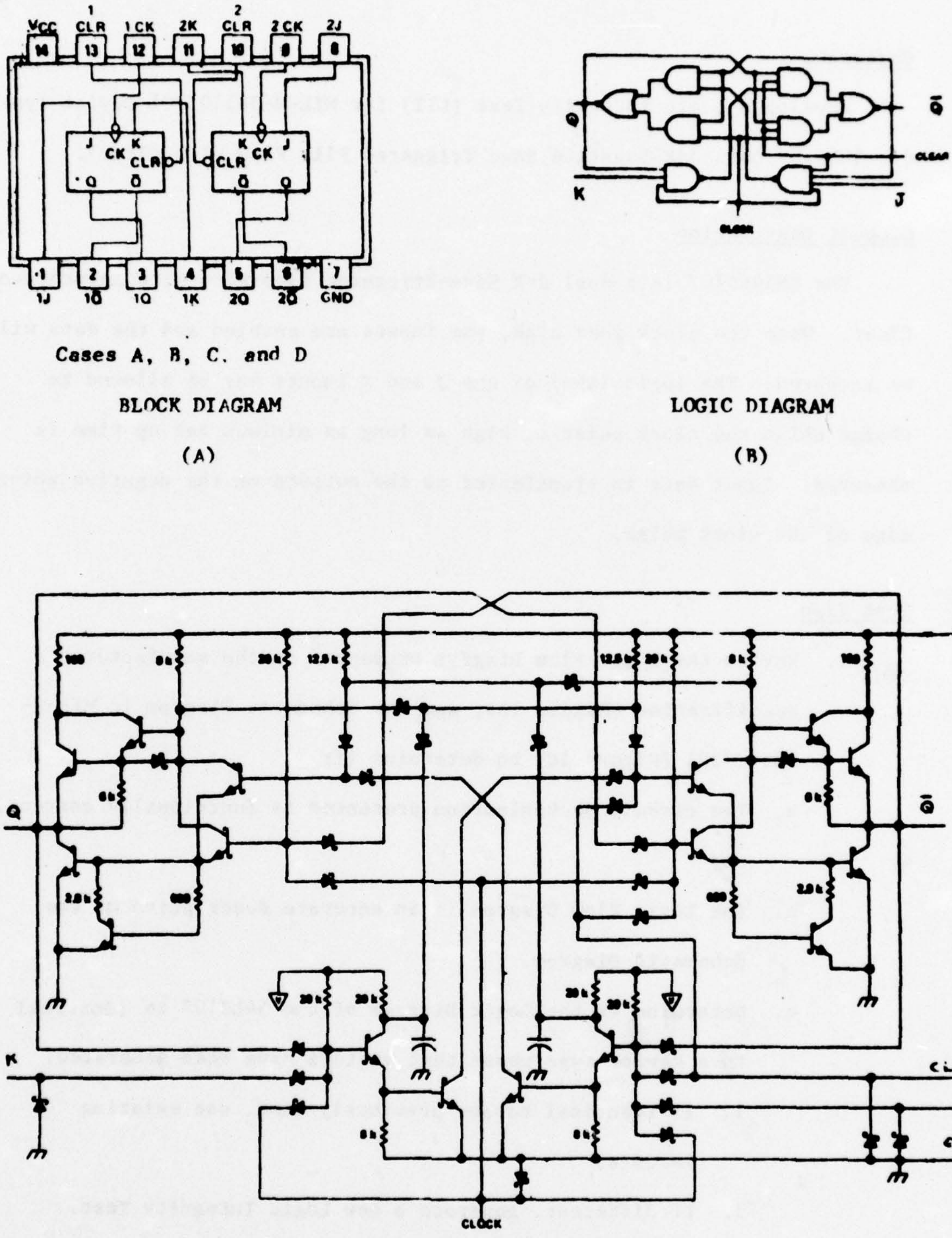
   b. The Logic Flow Diagram is an accurate description of the Schematic Diagram.

   c. Determine if the Logic Diagram of the 54LS107 is identical to a device type whose test vectors have been generated.

      1. If identical to any previously done, use existing vectors.

      2. If different, generate a new Logic Integritv Test.

II. Test a device with the vectors.



Cases A, B, C, and D

BLOCK DIAGRAM

(A)

LOGIC DIAGRAM

(B)

SCHEMATIC DIAGRAM

(C)

FIGURE 1

Summary

The Schematic Diagram and Logic Flow Diagram for the 54LS107 were reviewed. The circuit mechanization was found to be functionally correct. The Logic Flow Diagram requires the addition of two delays to operate properly and represent the true mechanization of the Schematic Diagram. The reasons for the addition of the delays is discussed in tie Analysis section.

It was determined that the 54LS107 has a fundamentally similar transistor diagram to the 54LS73. If the pin numbers of the 54LS73 were changed as in Table 1, then the 54LS73 is logically identical to the 54LS107.

Pin Names	Pin Numbers of 54LS73	Pin Numbers of 54LS107
1J	14	1
1$\overline{Q}$	13	2
1Q	12	3
1K	3	4
2Q	9	5
2$\overline{Q}$	8	6
GND	11	7
2J	7	8
2CLK	5	9
2CLR	6	10
2K	10	11
1CLK	1	12
1CLR	2	13
VCC	4	14

TABLE 1

Therefore, it is recommended that the 54LS107 be tested using the vectors developed for the 54LS73* (Table 2). These vectors provide a 100% TCL for both devices.

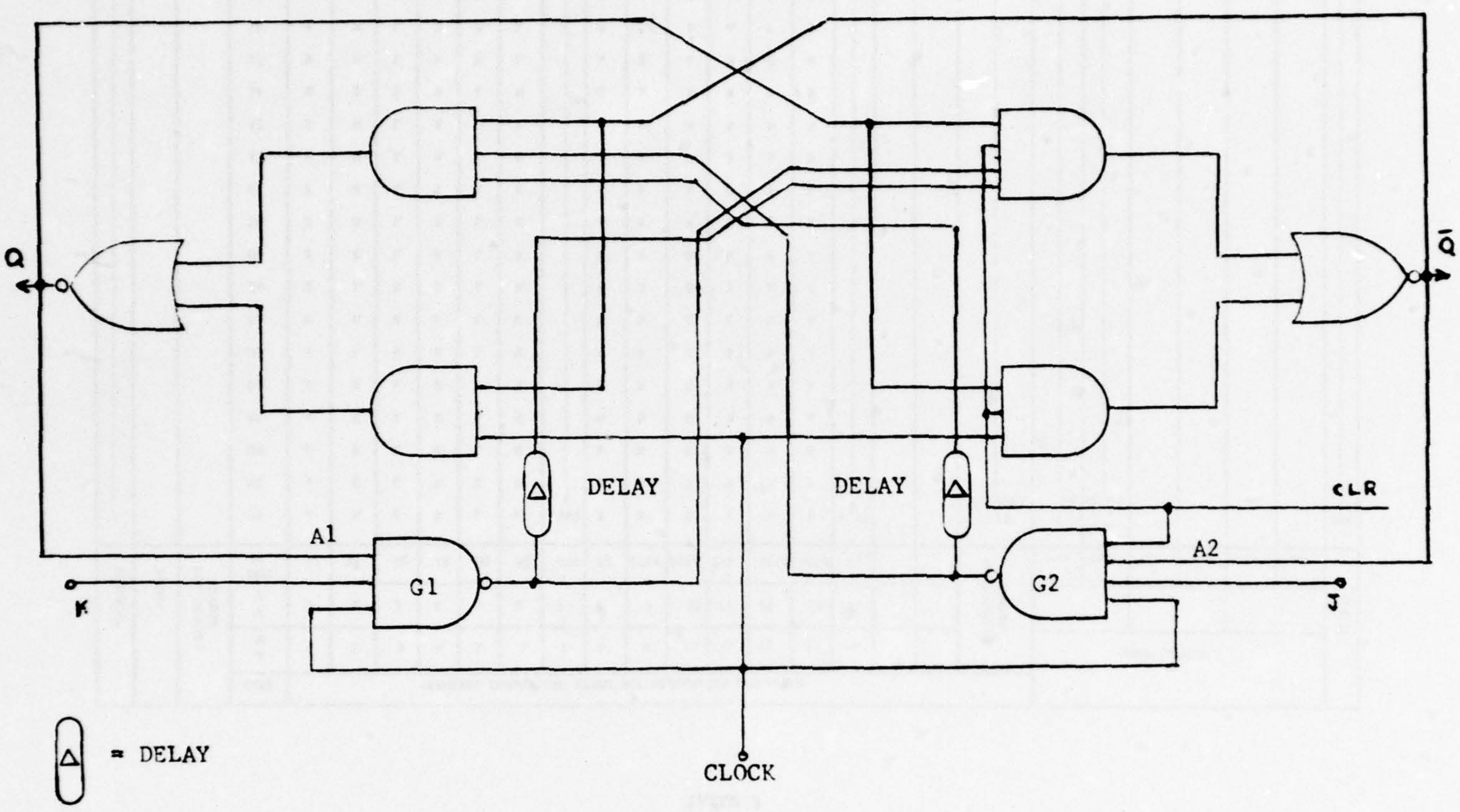
*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

A 54LS107 was successfully tested using these vectors.

Analysis

Analysis of the circuit operation of the 54LS107 indicated that proper functioning required relatively lqng propagation delays in gates G1 and G2 (Figure 2). These delays are necessary to allow input information from gates G1 and G2 to be accepted by the latch on a negative transition of the clock before the low level of the clock disables G1 and G2. Absence of these delays may result in an indeterminate state. These delays are incorporated in the Schematic Diagram. It is recommended that these delays be reflected in the Logic Diagram as presented in Figure 2.

There are two possible faults in each flip-flop which lead to an indeterminate state. These are S-A-1 faults on inputs A1 and A2 to gates G1 and G2 respectively (Figure 2). If either of these faults exists and the timing is such that the function is affected, the fault will be caught by the vectors presented in Table 2. If the internal gate delays are such that the fault does not affect the function, the fault is considered irrelevant. With this qualification, a TCL of 100% is achieved with tle vectors presented in Table 2.



RECOMMENDED LOGIC DIAGRAM

FIGURE 2

TABLE 2

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	1J	1Q̄	1Q	1K	2Q	2Q̄	GND	2J	2CLK	2CLR	2K	1CLK	1CLR	V_{CC}				Min	Max	Min	Max	Min	Max	
7			01	A	H	L	B	L	H	GND	B	B	B	B	B	B	4.5V			All Outputs	H or L as shown						VDC
			02	A	H	L	B	L	H		B	B	B	B	A	B											
			03	A	H	L	B	L	H		B	B	B	B	B	B											
			04	A	H	L	B	L	H		B	B	B	B	B	A											
			05	A	H	L	B	L	H		B	B	B	B	A	A											
			06	A	L	H	B	L	H		B	B	B	B	B	A											
			07	B	L	H	A	L	H		B	B	B	B	B	A											
			08	B	L	H	A	L	H		B	B	B	B	A	A											
			09	B	H	L	A	L	H		B	B	B	B	B	A											
			10	B	H	L	B	L	H		A	B	B	B	B	B											
			11	B	H	L	B	L	H		A	B	B	B	B	B											
			12	B	H	L	B	L	H		A	B	B	B	B	B											
			13	B	H	L	B	L	H		A	B	A	B	B	B											
			14	B	H	L	B	L	H		A	A	A	B	B	B											
			15	B	H	L	B	H	L		A	B	A	B	B	B											
			16	B	H	L	B	H	L		B	B	A	A	B	B											
			17	B	H	L	B	H	L		B	A	A	A	B	B											
			18	B	H	L	B	L	H		B	B	A	A	B	B											
			19	B	H	L	B	L	H		B	B	B	B	B	B											
7			20	B	H	L	B	L	H	GND	B	B	A	B	B	A	4.5V			All Outputs	H or L as shown						VDC

TABLE 2 (Cont'd)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	1J	$1\overline{Q}$	1Q	1K	2Q	$2\overline{Q}$	GND	2J	2CLK	2CLR	2K	1CLK	1CLR	V_{CC}				Min	Max	Min	Max	Min	Max	
7			21	B	H	L	B	L	H	GND	B	A	A	B	A	A	4.5V			All Outputs	H or L as shown						VDC
			22	B	H	L	B	L	H		B	B	A	B	B	A											
			23	A	H	L	A	L	H		A	B	A	A	B	A											
			24	A	H	L	A	L	H		A	A	A	A	A	A											
			25	A	L	H	A	H	L		A	B	A	A	B	A											
			26	B	L	H	B	H	L		B	B	A	B	B	A											
			27	B	L	H	B	H	L		B	A	A	B	A	A											
			28	B	L	H	B	H	L		B	B	A	B	B	A											
			29	A	L	H	A	H	L		A	B	A	A	B	A											
			30	A	L	H	A	H	L		A	A	A	A	A	A											
7			31	A	H	L	A	L	H	GND	A	B	A	A	B	A	4.5V			All Outputs	H or L as shown						VDC

NOTES:

1. L = 0.4V
   H = 2.4V

2. A = 2.0V min.
   B = 0.7V max.

3. Test must be run in sequence shown.

REPORT NO. 5.5

Logic Integrity Test of the 54LS109 (Vendor C) Dual J-K Edge-Triggered Flip-Flop

Objective:

Develop A Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 09 (54LS109 Dual J-K Edge-Triggered Flip-Flops).

Circuit Description:

The SN54LS109 is a dual J-$\overline{K}$ Edge-Triggered Flip-Flop with separate asynchronous Preset and Clear inputs. The clocking operation is independent of rise and fall times of the clock waveform. The J-$\overline{K}$ design allows operation as a D type Flip-Flop by simply connecting the J and $\overline{K}$ pins together. Input data is transferred tothe outputs on the positive going edge of the clock pulse.

Test Plan:

I. Review the Logic Flow Diagram presented in the manufacturer's specification (Figure 1B) and the Schematic Diagram in MIL-M-38510/301 (Figure 1C) to determine if:

   a. The circuit mechanization presented is functionally correct, and,

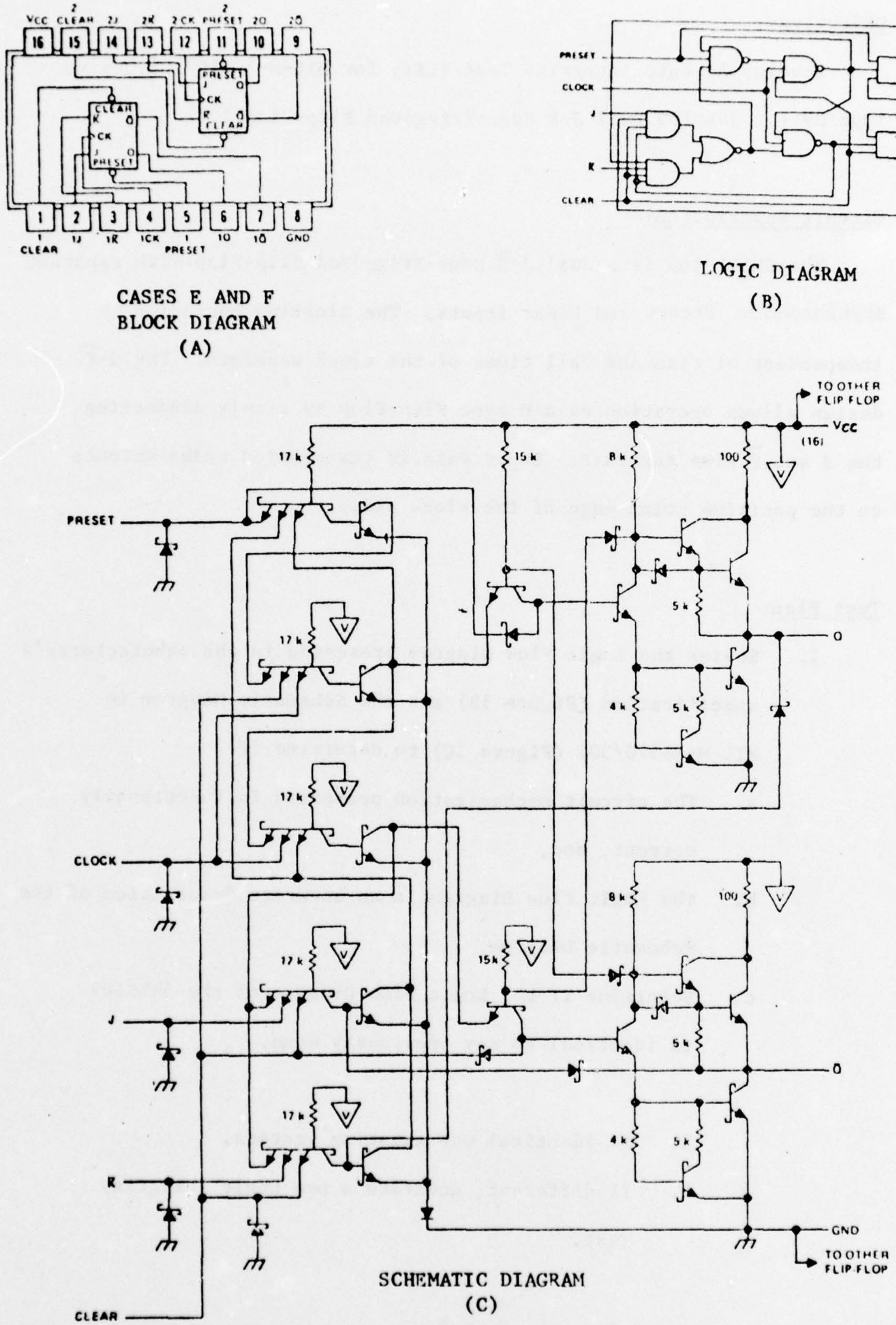
   b. the Logic Flow Diagram is an accurate description of the Schematic Diagram.

   c. Determine if the Logic Flow Diagram of the 54LS109 is identical to any previously done.

      1. If identical use existing vectors.

      2. If different, generate a new Logic Integrity Test.

II. Test a device with the vectors.



CASES E AND F
BLOCK DIAGRAM
(A)

LOGIC DIAGRAM
(B)

SCHEMATIC DIAGRAM
(C)

FIGURE I

Summary:

The Schematic Diagram and logic Flow Diagram for the 54LS109 were reviewed. The circuit mechanization was found to be functionally correct.

Since the 54LS109 was not identical to any previously done, a set of test vectors was generated by GEOS. Based on the testing criteria*, of the Fault Analysis Simulation (FAS) Program, the complete set of test vectors, presented on Table 1, have a TCL of 100%.

A 54LS109 was successfully tested using these vectors.

* Relative to the FAS Program, the number of tests needed to check a device completely is a S-A-1 and S-A-0 test on the output lead and the unique test for each input lead on each gate in the network. This is equivalent to testing all leads for S-A-1 and S-A-0.

TABLE 1

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																	TEST LIMITS						UNITS
			E	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	MEASUREMENT TERMINAL							
			F	1	2	3	4	5	6	7.	8	9	10	11	12	13	14	15	16								
			TEST NO.	1CLR	1J	$1\overline{K}$	1CLK	1PR	1Q	$1\overline{Q}$	GND	$2\overline{Q}$	2Q	2PR	2CLK	$2\overline{K}$	2J	2CLR	V_{cc}		Min	Max	Min	Max	Min	Max	
7			1	A	A	A	B	B	H	L	GND	L	H	B	B	A	A	A	4.5V	ALL OUTPUTS	H or As	L shown					VDC
			2	A	A	A	B	A	H	L		L	H	A	B	A	A	A									
			3	A	A	A	A	A	H	L		L	H	A	A	A	A	A									
			4	A	A	B	A	A	H	L		L	H	A	A	B	A	A									
			5	A	A	B	B	A	H	L		L	H	A	B	B	A	A									
			6	A	A	B	A	A	L	H		H	L	A	A	B	A	A									
			7	A	A	B	B	A	L	H		H	L	A	B	B	A	A									
			8	B	A	B	B	A	L	H		H	L	A	B	B	A	B									
			9	A	A	B	B	A	L	H		H	L	A	B	B	A	A									
			10	A	A	B	A	A	H	L		L	H	A	A	B	A	A									
			11	A	A	B	B	A	H	L		L	H	A	B	B	A	A									
			12	A	A	B	A	A	L	H		H	L	A	A	B	A	A									
			13	A	A	B	B	A	L	H		H	L	A	B	B	A	A									
			14	A	A	B	B	B	H	L		L	H	B	B	B	A	A									
			15	A	A	B	A	B	H	L		L	H	B	A	B	A	A									
			16	A	A	B	A	A	H	L		L	H	A	A	B	A	A									
			17	A	A	B	B	A	H	L		L	H	A	B	B	A	A									
7			18	A	A	B	A	A	L	H	GND	H	L	A	A	B	A	A	4.5V	ALL OUTPUTS	H or As	L shown					VDC

T A B L E 1 (Continued)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			E	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	1CLR	1J	$1\overline{K}$	1CLK	1PR	1Q	$1\overline{Q}$	GND	$2\overline{Q}$	2Q	2PR	2CLK	$2\overline{K}$	2J	2CLR	V_{cc}		Min	Max	Min	Max	Min	Max	
7			19	A	A	B	B	A	L	H	GND	H	L	A	B	B	A	A	4.5V	ALL OUTPUTS	H as	or L shown					VDC
			20	B	B	B	B	A	L	H		H	L	A	B	B	B	B									
			21	A	B	B	B	A	L	H		H	L	A	B	B	B	A									
			22	A	B	B	A	A	L	H		H	L	A	A	B	B	A									
			23	A	A	B	A	A	L	H		H	L	A	A	B	A	A									
			24	A	A	B	B	A	L	H		H	L	A	B	B	A	A									
			25	A	A	B	B	B	H	L		L	H	B	B	B	A	A									
			26	A	A	B	B	A	H	L		L	H	A	B	B	A	A									
			27	A	A	B	A	A	L	H		H	L	A	A	B	A	A									
			28	A	A	B	B	A	L	H		H	L	A	B	B	A	A									
			29	B	A	B	B	A	L	H		H	L	A	B	B	A	B									
			30	B	A	B	A	A	L	H		H	L	A	A	B	A	B									
			31	A	A	B	A	A	L	H		H	L	A	A	B	A	A									
			32	A	A	B	B	A	L	H		H	L	A	B	B	A	A									
			33	A	B	A	B	A	L	H		H	L	A	B	A	B	A									
			34	A	B	A	A	A	L	H		H	L	A	A	A	B	A									
			35	A	A	A	A	B	H	L		L	H	B	A	A	A	A									
7			36	A	A	A	A	A	H	L	GND	L	H	A	A	A	A	A	4.5V	ALL OUTPUTS	H as	or L shown					VDC

TABLE 1 (Continued)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			E	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	1CLR	1J	$\overline{1K}$	1CLK	1PR	1Q	$\overline{1Q}$	GND	$\overline{2Q}$	2Q	2PR	2CLK	$\overline{2K}$	2J	2CLR	V_{cc}		Min	Max	Min	Max	Min	Max	
7			37	B	A	A	A	A	L	H	GND	H	L	A	A	A	A	B	4.5V	ALL OUTPUTS	H As	or L shown					VDC
↓			38	A	B	B	B	B	H	L	↓	L	H	B	B	B	B	A	↓	↓	↓	↓					↓
↓			39	A	B	B	B	A	H	L	↓	L	H	A	B	B	B	A	↓	↓	↓	↓					↓
7			40	B	B	B	B	A	L	H	GND	H	L	A	B	B	B	B	4.5V	ALL OUTPUTS	H As	or L shown					VDC

NOTES:

1. L = 0.4V Max. H = 2.4V Min.
2. A = 2.0V Min. B = 0.7V Max.
3. TEST must be run in sequence shown.

REPORT NO. 5.6

Logic Integrity Test of the 54LS112 (Vendor C) Dual J-K Edge-Triggered Flip-Flop

Objective:

Develop A Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 03 (54LS112 Dual J-K Edge Triggered Flip Flops).

Circuit Description:

The SN54LS112 is a dual J-K Edge-Triggered Flip-Flop with separate Asynchronous Preset and Clear inputs. When the clock goes high, the inputs are enabled and the data will be accepted. The logic level of the J and K inputs may be allowed to change while the clock pulse is high as long as minimum set up and hold times are observed. Input data is transferred to the outputs on the negative going edge of the clock pulse.

Test Plan:

I. Review the Logic Flow Diagram presented in the manufacturer's specification (Figure 1B), and the Schematic Diagram in MIL-M-38510/301 (Figure 1C) to determine if:

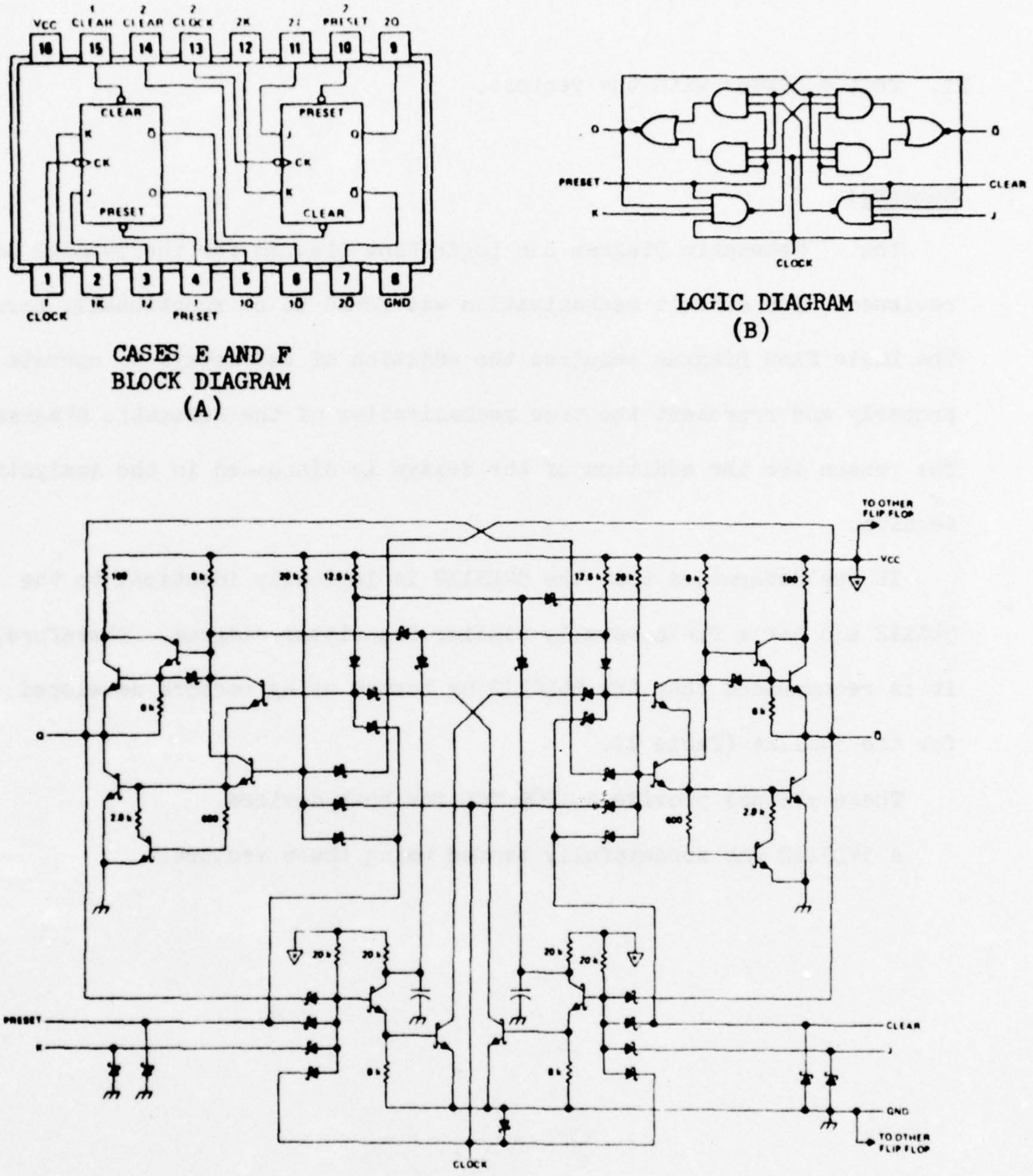
a. The circuit mechanization presented is functionally correct, and,

b. the Logic Flow Diagram is an accurate description of the Schematic Diagram.

c. Determine if the Logic Flow Diagram of the 54LS112 is identical to the 54S112.

1. If identical use 54S112 vectors

2. If different, generate a new Logic Integrity Test



CASES E AND F
BLOCK DIAGRAM
(A)

LOGIC DIAGRAM
(B)

SCHEMATIC DIAGRAM
(C)

FIGURE 1

II. Test a device with the vectors.

Summary:

The Schematic Diagram and Logic Flow Diagram for the 54LS112 were reviewed. The circuit mechanization was found to be functionally correct. The Logic Flow Diagram requires the addition of two delays to operate properly and represent the true mechanization of the Schematic Diagram. The reason for the addition of the delays is discussed in the Analysis section.

It was determined that the 54LS112 is logically identical to the 54S112 and has a fundamentally similar transistor diagram. Therefore, it is recommended that the 54LS112 be tested using vectors developed for the 54S112* (Table 1).

These vectors provide a 100% TCL for both devices.

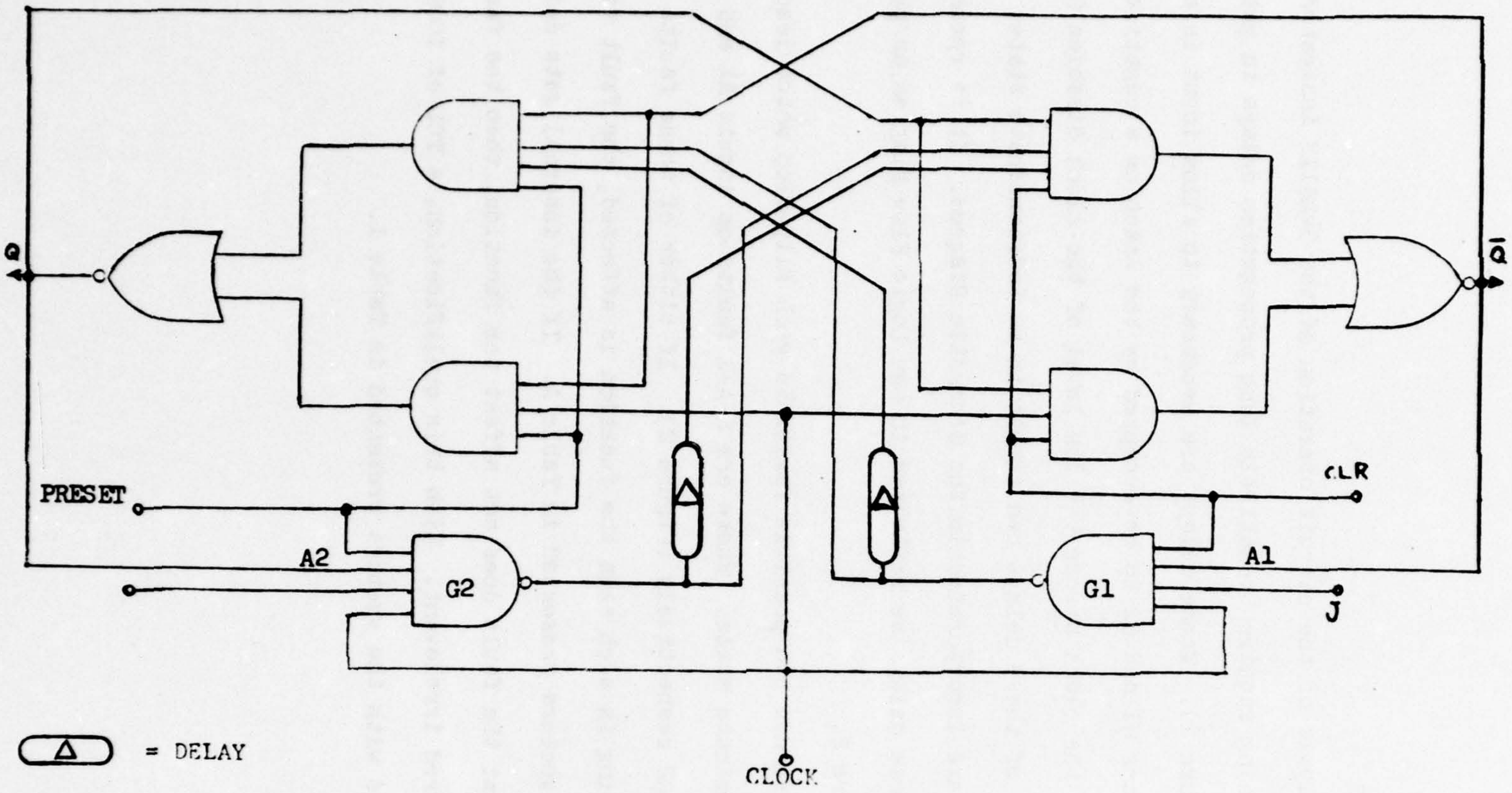
A 54LS112 was successfully tested using these vectors.

*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

Analysis

Analysis of the circuit operation of the 54LS112 indicated that proper functioning requires relatively long propagation delays in gates G1 and G2 (Figure 2). These delays are necessary to allow input information from gates G1 and G2 to be accepted by the latch on a negative transition of the clock before the low level of the clock disables G1 and G2. Absence of these delays can result in an indeterminate state. These delays are incorporated in the Schematic Diagram. It is recommended that these delays be reflected in the Logic Flow Diagram as presented in Figure 2.

There are two possible faults in each flip-flop which lead to an indeterminate state. These are S-A-1 faults on inputs A1 and A2 to gates G1 and G2 respectively (Figure 2). If either of these faults exists, and the timing is such that the function is affected, the fault will be caught by the vectors presented in Table 1. If the internal gate delays are such that the fault does not affect the function, then the fault is considered irrelevant. With this qualification, a TCL of 100% is achieved with the vectors presented in Table 1.



RECOMMENDED LOGIC DIAGRAM

FIGURE 2

TABLE 1

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	1CK	1K	1J	1PR	1Q	$1\overline{Q}$	$2\overline{Q}$	GND	2Q	2PR	2J	2K	2CK	2CLR	1CLR	Vcc		Min	Max	Min	Max	Min	Max	
7			1	B	B	A	A	L	H	H	GND	L	A	A	B	B	B	B	4.5V	ALL Outputs	H or L as shown						VDC
			2	A	B	A	A	L	H	H		L	A	A	B	A	B	B									
			3	B	B	A	A	L	H	H		L	A	A	B	B	B	B									
			4	B	A	B	B	H	L	L		H	B	B	A	B	A	A									
			5	A	A	B	B	H	L	L		H	B	B	A	A	A	A									
			6	B	A	B	B	H	L	L		H	B	B	A	B	A	A									
			7	B	B	B	A	L	H	H		L	A	B	B	B	B	B									
			8	B	B	B	A	L	H	H		L	A	B	B	B	A	A									
			9	A	B	B	A	L	H	H		L	A	B	B	A	A	A									
			10	B	B	B	A	L	H	H		L	A	B	B	B	A	A									
			11	B	B	B	B	H	L	L		H	B	B	B	B	A	A									
			12	B	B	B	A	H	L	L		H	A	B	B	B	A	A									
			13	A	B	B	A	H	L	L		H	A	B	B	A	A	A									
			14	B	B	B	A	H	L	L		H	A	B	B	B	A	A									
			15	B	B	A	A	L	H	H		L	A	A	B	B	B	B									
			16	B	B	A	A	L	H	H		L	A	A	B	B	A	A									
			17	A	B	A	A	L	H	H		L	A	A	B	A	A	A									
			18	B	B	A	A	H	L	L		H	A	A	B	B	A	A									
			19	B	A	B	B	H	L	L		H	B	B	A	B	A	A									
7			20	B	A	B	A	H	L	L	GND	H	A	B	A	B	A	A	4.5V	ALL Outputs	H or L as shown						VDC

TABLE 1 (Continued)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			E	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	1CK	1K	1J	1PR	1Q	1Q̄	2Q̄	GND	2Q	2PR	2J	2K	2CK	2CLR	1CLR	Vcc		Min	Max	Min	Max	Min	Max	
7			21	A	A	B	A	H	L	L	GND	H	A	B	A	A	A	A	4.5V	All Outputs	H or L as shown						VDC
↓			22	B	A	B	A	L	H	H	↓	L	A	B	A	B	A	A	↓	↓	↓	↓					↓
↓			23	A	A	A	B	H	H	H	↓	H	B	A	A	A	B	B	↓	↓	↓	↓					↓
↓			24	B	A	A	A	L	H	H	↓	L	A	A	A	B	B	B	↓	↓	↓	↓					↓
↓			25	B	A	A	A	L	H	H	↓	L	A	A	A	B	A	A	↓	↓	↓	↓					↓
↓			26	A	A	A	A	L	H	H	↓	L	A	A	A	A	A	A	↓	↓	↓	↓					↓
↓			27	B	A	A	A	H	L	L	↓	H	A	A	A	B	A	A	↓	↓	↓	↓					↓
↓			28	A	A	A	A	H	L	L	↓	H	A	A	A	A	A	A	↓	↓	↓	↓					↓
7			29	B	A	A	A	L	H	H	GND	L	A	A	A	B	A	A	4.5V	All Outputs	H or L as shown						VDC

NOTES:

1. L = 0.4V max. H = 2.4V min.
2. A = 2.0V min. B = 0.7V max.
3. Test must be run in sequence shown.

REPORT NO. 5.7

Logic Integrity Test for the 54LS113 (Vendor C) Dual J-K Edge-Triggered Flip-Flop

Objective:

Develop A Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 04 (54LS113 Dual J-K Edge Triggered Flip Flop).

Circuit Description

The SN54LS113 is a dual J-K Edge-Triggered Flip-Flop with Asynchronous Presets. When the clock goes high, the inputs are enabled and the data will be accepted. The logic level of the J and K inputs may be allowed to change while the clock pulse is high as long as minimum set up time is observed. Input data is transferred to the outputs on the negative going edge of the clock pulse.

Test Plan

I. Review the Logic Flow Diagram presented in the Manufacturer's Specification (Figure 1B), and the Schematic Diagram in MIL-M-38510/301 (Figure 1C) to determine if:

a. The circuit mechanization presented is functionally correct, and

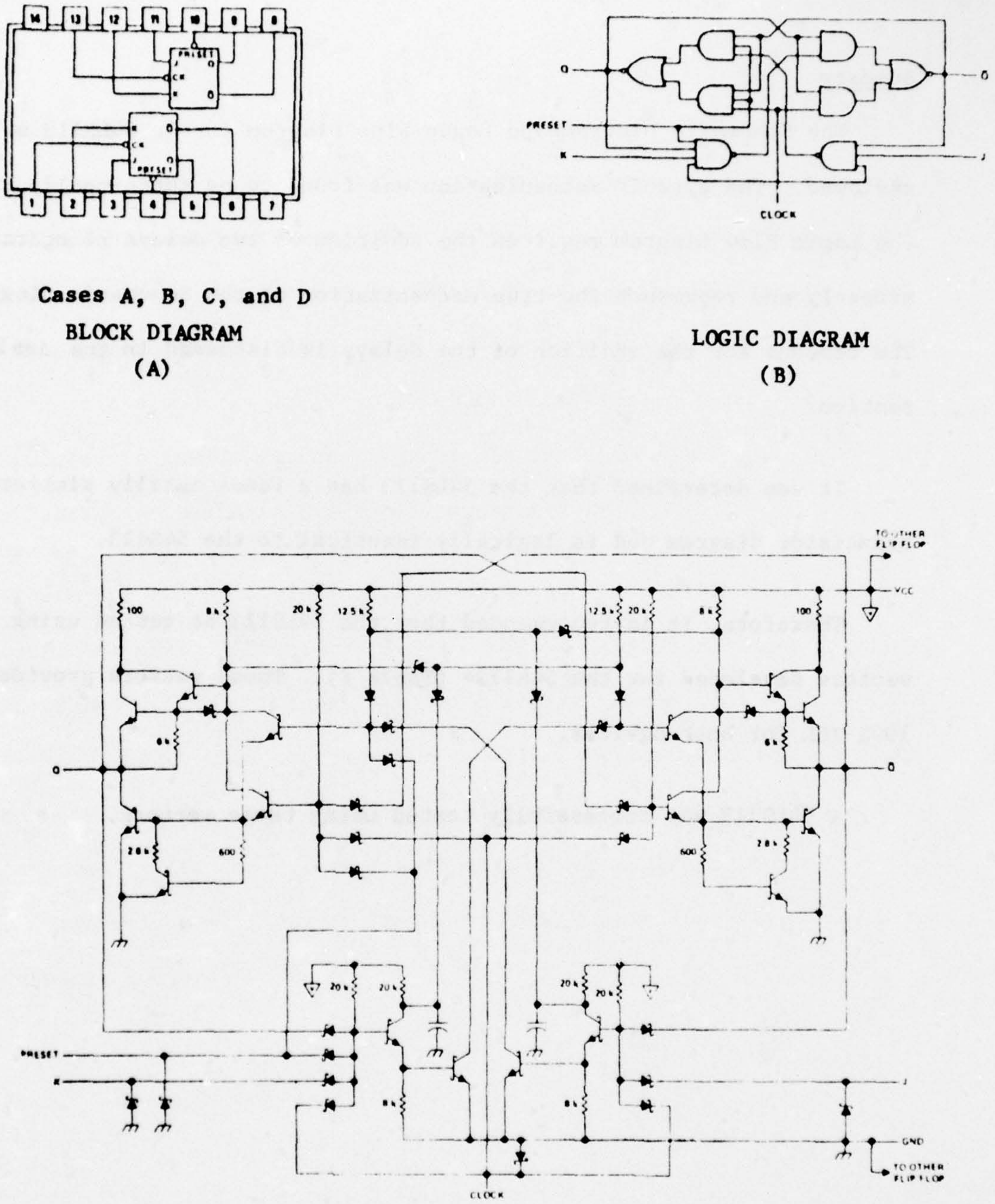
b. The Logic Flow Diagram is an accurate description of the Schematic Diagram.

c. Determine if Logic Flow Diagram of the 54LS113 is identical to that of the 54S113.

1. If identical, use existing vectors.

2. If different, generate a new Logic Integrity Test.

II. Test a device with the vectors.



Cases A, B, C, and D

BLOCK DIAGRAM

(A)

LOGIC DIAGRAM

(B)

SCHEMATIC DIAGRAM

(C)

FIGURE 1

Summary

The Schematic Diagram and Logic Flow Diagram for he 54LS113 were reviewed. The circuit mechanization was found to be functionally correct. The Logic Flow Diagram requires the addition of two delays to operate properly and represent the true mechanization of the Schematic Diagram. The reasons for the addition of the delays is discussed in the Analysis section.

It was determined that the 54LS113 has a fundamentally similar transistor diagram and is logically identical to the 54S113.

Therefore, it is recommended that the 54LS113 be tested using the vectors developed for the 54S113* (Table I). These vectors provide a 100% TCL for both devices.

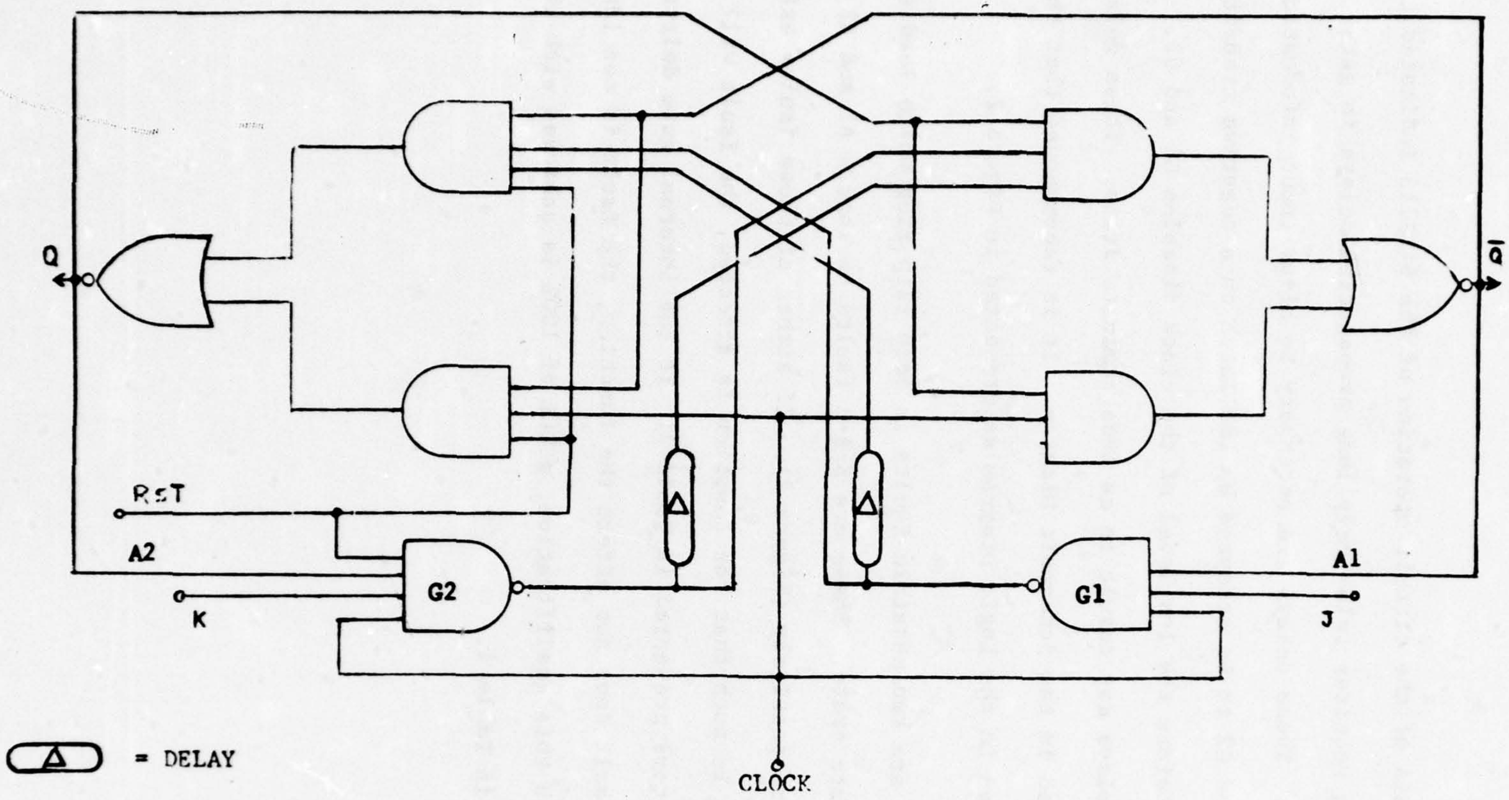
A 54LS113 was successfully tested using these vectors.

*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

Analysis

Analysis of the circuit operation of the 54LS113 indicated that proper functioning requires relatively long propagation delays in gates G1 and G2 (Figure 2). These delays are necessary to allow input information from gates G1 and G2 to be accepted by the latch on a negative transition of the clock before the low level of the clock disables G1 and G2. Absence of these delays may result in an indeterminate state. These delays are incorporated in the Schematic Diagram. It is recommended that these delays be reflected in the Logic Diagram as presented in Figure 2.

There are two possible faults in each flip-flop which lead to an indeterminate state. These are S-A-1 faults on inputs A1 and A2 to gates G1 and G2 respectively (Figure 2). If either of these faults exists and the timing is such that the function is affected, the fault will be caught by the vectors presented in Table 1. If the internal gate delays are such that the fault does not affect the function, the fault is considered irrelevant. With this qualification, a TCL of 100% is achieved with the vectors presented in Table 1.



RECOMMENDED LOGIC DIAGRAM

FIGURE 2

Table 1

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14			MEASUREMENT TERMINAL							
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	1CK	1K	1J	1PR	1Q	$1\bar{Q}$	GND	$2\bar{Q}$	2Q	2PR	2J	2K	2CK	Vcc				Min	Max	Min	Max	Min	Max	
7			1	B	A	B	B	H	L	GND	L	H	B	B	B	B	4.5V			All Outputs	H or L as shown						VDC
			2	A	A	B	B	H	L		L	H	B	B	B	B											
			3	B	A	B	B	H	L		L	H	B	B	B	B											
			4	B	A	B	A	H	L		L	H	B	B	B	B											
			5	A	A	B	A	H	L		L	H	B	B	B	B											
			6	B	A	B	A	L	H		L	H	B	B	B	B											
			7	B	B	A	A	L	H		L	H	B	B	B	B											
			8	A	B	A	A	L	H		L	H	B	B	B	B											
			9	B	B	A	A	H	L		L	H	B	B	B	B											
			10	B	B	B	B	H	L		L	H	B	B	A	B											
			11	B	B	B	B	H	L		L	H	B	B	A	B											
			12	B	B	B	B	H	L		L	H	B	B	A	B											
			13	B	B	B	B	H	L		L	H	A	B	A	B											
			14	B	B	B	B	H	L		L	H	A	B	A	A											
			15	B	B	B	B	H	L		H	L	A	B	A	B											
			16	B	B	B	B	H	L		H	L	A	A	B	B											
			17	B	B	B	B	H	L		H	L	A	A	B	A											
			18	B	B	B	B	H	L		L	H	A	A	B	B											
			19	B	B	B	B	H	L		L	H	B	B	B	B											
7			20	B	B	B	A	H	L	GND	L	H	A	B	B	B	4.5V			All Outputs	H or L as shown						VDC

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	1CK	1K	1J	1PR	1Q	$\overline{1Q}$	GND	$\overline{2Q}$	2Q	2PR	2J	2K	2CK	Vcc				Min	Max	Min	Max	Min	Max	
7			21	A	B	B	A	H	L	GND	L	H	A	B	B	A	4.5V			All Outputs	H or L as shown						VDC
↓			22	B	B	B	A	H	L	↓	L	H	A	B	B	B	↓			↓	↓	↓					↓
↓			23	B	A	A	A	H	L	↓	L	H	A	A	A	B	↓			↓	↓	↓					↓
↓			24	A	A	A	A	H	L	↓	L	H	A	A	A	A	↓			↓	↓	↓					↓
↓			25	B	A	A	A	L	H	↓	H	L	A	A	A	B	↓			↓	↓	↓					↓
↓			26	B	B	B	A	L	H	↓	H	L	A	B	B	B	↓			↓	↓	↓					↓
↓			27	A	B	B	A	L	H	↓	H	L	A	B	B	A	↓			↓	↓	↓					↓
↓			28	B	B	B	A	L	H	↓	H	L	A	B	B	B	↓			↓	↓	↓					↓
↓			29	B	A	A	A	L	H	↓	H	L	A	A	A	B	↓			↓	↓	↓					↓
↓			30	A	A	A	A	L	H	↓	H	L	A	A	A	A	↓			↓	↓	↓					↓
7			31	B	A	A	A	H	L	GND	L	H	A	A	A	B	4.5V			All Outputs	H or L as shown						VDC

NOTES:

1. L = 0.4V max. H = 2.4V min.
2. A = 2.0V min. B = 0.7V max.
3. Test must be run in sequence shown.

REPORT NO. 5.8

Logic Integrity Test for the 54LS114 (Vendor C) Dual J-K Edge-Triggered Flip-Flop

Objective:

Develop a Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 05 (54LS114 Dual J-K Edge Triggered Flip Flop).

Circuit Description

The SN54LS114 is a dual J-K Edge-Triggered Flip-Flop with common clock, common clear direct, and separate set direct inputs. When the clock goes high, the inputs are enabled and the data will be accepted. The logic level of the J and K inputs may be allowed to change while the clock pulse is high as long as minimum set up time is observed. Input data is transferred to the outputs on the negative going edge of the clock pulse.

Test Plan

I. Review the Logic Flow Diagram presented in the manufacturer's specification (Figure 1B), and the Schematic Diagram in MIL-M38510/301 (Figure 1C) to determine if:

   a. The circuit mechanization presented is functionally correct, and

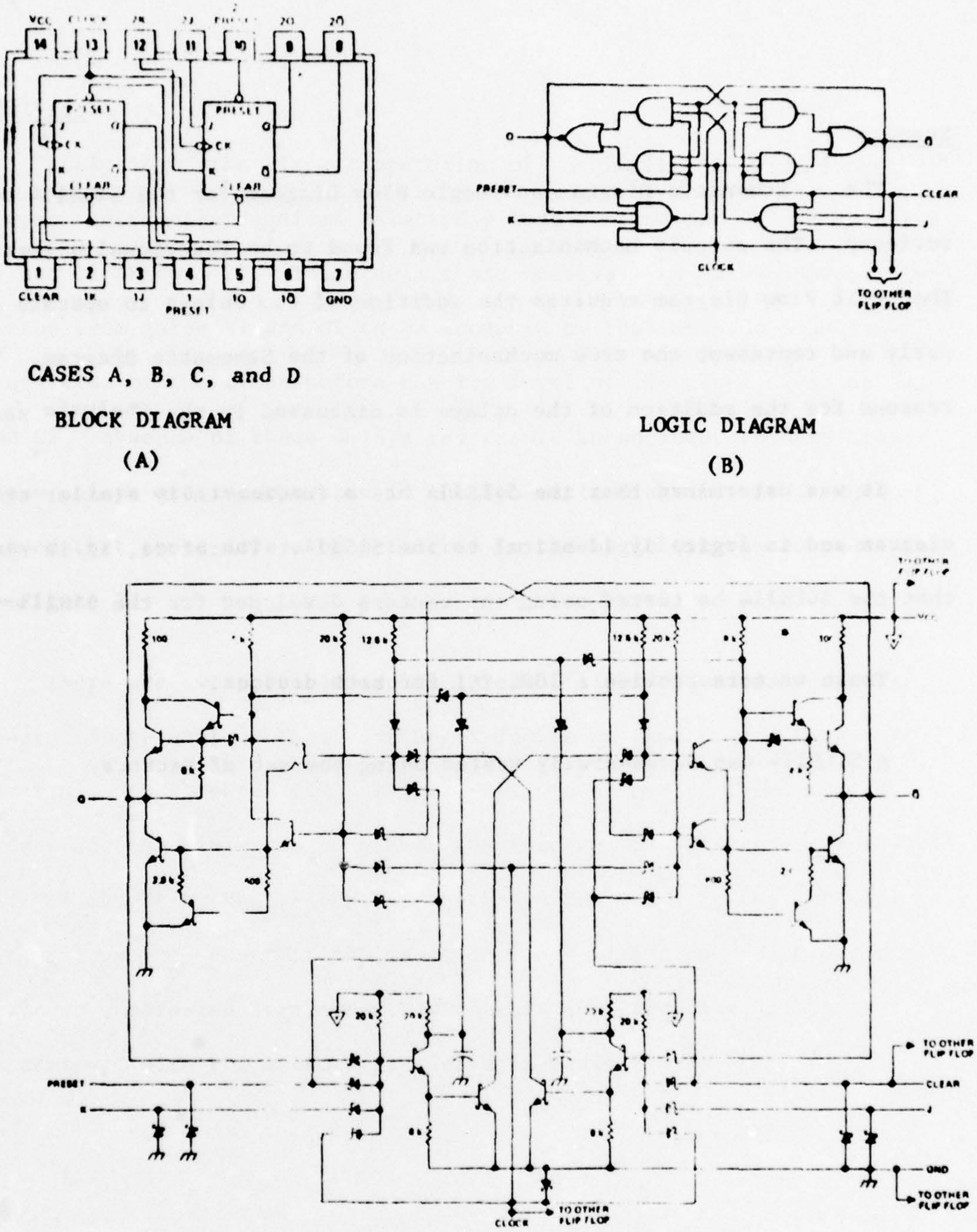
   b. The Logic Flow Diagram is an accurate description of the Schematic Diagram.

   c. Determine if the Logic Flow Diagram of the 54LS114 is identical to the 54S114.

      1. If identical, use the 54S114 vectors.

      2. If different, generate a new Logic Integrity Test.

II. Test a device with the vectors.



CASES A, B, C, and D

BLOCK DIAGRAM

(A)

LOGIC DIAGRAM

(B)

SCHEMATIC DIAGRAM

(C)

FIGURE 1

Summary

The Schematic Diagram and Logic Flow Diagram for the 54LS114 were reviewed. The circuit mechanization was found to be functionally correct. The Logic Flow Diagram requires the addition of two delays to operate properly and represent the true mechanization of the Schematic Diagram. The reasons for the addition of the delays is discussed in the Analysis section.

It was determined that the 54LS114 has a fundamentally similar transistor diagram and is logically identical to the 54S114. Therefore, it is recommended that the 54LS114 be tested using the vectors developed for the 54S114* (Table 1).

These vectors provide a 100% TCL for both devices.

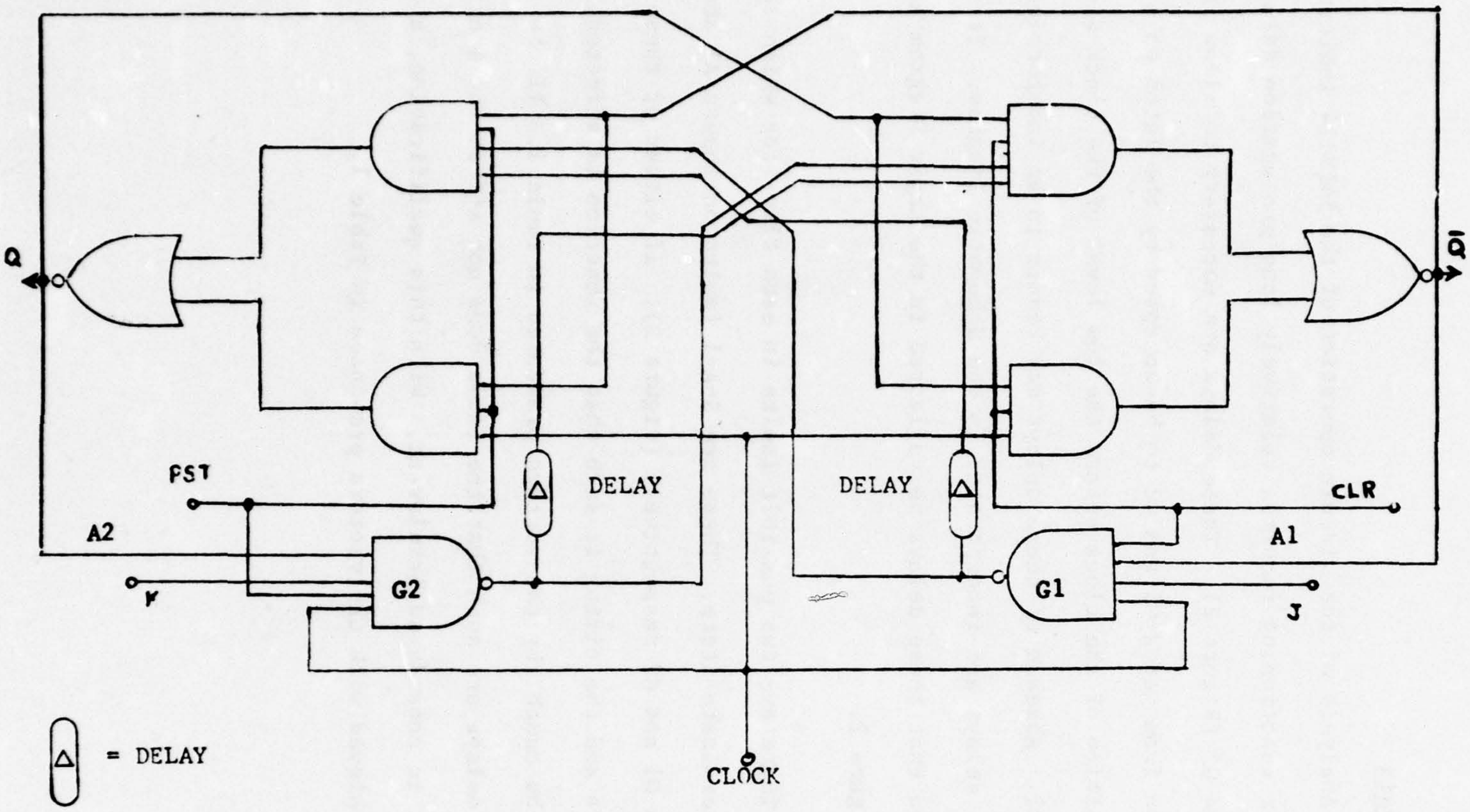
A 54LS114 was successfully tested using the set of vectors.

*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

Analysis

Analysis of the circuit operation of the 54LS114 indicated that proper functioning requires relatively long propagation delays in gates G1 and G2 (Figure 2). These delays are necessary to allow input information from gates G1 and G2 to be accepted by the latch on a negative transition of the clock before the low level of the clock disables G1 and G2. Absence of these delays may result in an indeterminate state. These delays are incorporated in the Schematic Diagram. It is recommended that these delays be reflected in the Logic Diagram as presented in Figure 2.

There are two possible faults in each flip-flop which lead to an indeterminate state. These are S-A-1 faults on inputs A1 and A2 to gates G1 and G2 respectively (Figure 2). If either of these faults exists and the timing is such that the function is affected, the fault will be caught by the vectors presented in Table 1. If the internal gate delays are such that the fault does not affect the function, the fault is considered irrelevant. With this qualification, a TCL of 100% is achieved with the vectors presented in Table 1.



RECOMMENDED LOGIC DIAGRAM

FIGURE 2

TABLE 1

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	CLR	1K	1J	1PR	1Q	$1\overline{Q}$	GND	$2\overline{Q}$	2Q	2PR	2J	2K	CK	VCC				Min	Max	Min	Max	Min	Max	
7			1	B	B	A	A	L	H	GND	H	L	A	A	B	A	4.5V			All Outputs	H or L as shown						VDC
			2	B	B	A	A	L	H		H	L	A	A	B	B											
			3	B	B	A	A	L	H		H	L	A	A	B	A											
			4	A	B	A	A	L	H		L	H	B	B	B	A											
			5	A	B	A	A	H	L		L	H	B	B	B	B											
			6	A	B	A	A	H	L		L	H	B	B	B	A											
			7	A	B	B	A	H	L		L	H	B	B	B	A											
			8	A	B	B	A	H	L		L	H	B	B	B	B											
			9	A	B	B	A	B	L		L	H	B	B	B	A											
			10	A	A	B	A	H	L		L	H	B	B	B	A											
			11	A	A	B	A	L	H		L	H	B	B	B	B											
			12	A	A	B	A	L	H		L	H	B	B	B	A											
			13	A	A	B	B	H	L		L	H	B	B	B	A											
			14	A	A	B	B	H	L		L	H	B	B	B	B											
			15	B	B	B	A	L	H		H	L	A	A	B	A											
			16	A	B	B	B	H	L		H	L	A	A	B	A											
			17	A	B	B	B	H	L		L	H	A	A	B	B											
			18	A	B	B	B	H	L		L	H	A	A	B	A											
			19	A	B	B	B	H	L		L	H	A	B	B	A											
7			20	A	B	B	B	H	L	GND	L	H	A	B	B	B	4.5V			All Outputs	H or L as shown						VDC

TABLE 1 (con't.)

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			A B	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			C D	1	2	3	4	5	6	7	8	9	10	11	12	13	14										
			TEST NO.	CLR	1K	1J	1PR	1Q	1Q̄	GND	2Q̄	2Q	2PR	2J	2K	CK	VCC				Min	Max	Min	Max	Min	Max	
7			21	A	B	B	B	H	L	GND	L	H	A	B	B	A	4.5V			All Outputs	H or L as shown						VDC
			22	A	B	B	B	H	L		L	H	A	B	A	A											
			23	A	B	B	B	H	L		H	L	A	B	A	B											
			24	A	B	B	B	H	L		H	L	A	B	A	A											
			25	A	B	B	B	H	L		L	H	B	B	A	A											
			26	A	B	B	B	H	L		L	H	B	B	A	B											
			27	B	A	A	B	H	H		H	H	B	A	A	A											
			28	B	B	B	A	L	H		H	L	A	B	B	A											
			29	A	B	B	A	L	H		H	L	A	B	B	A											
			30	A	B	B	A	L	H		H	L	A	B	B	B											
			31	A	A	A	A	L	H		H	L	A	A	A	B											
			32	A	A	A	A	L	H		H	L	A	A	A	A											
			33	A	A	A	A	H	L		L	H	A	A	A	B											
			34	A	B	B	A	H	L		L	H	A	B	B	B											
			35	A	B	B	A	H	L		L	H	A	B	B	A											
			36	A	B	B	A	H	L		L	H	A	B	B	B											
			37	A	A	A	A	H	L		L	H	A	A	A	B											
			38	A	A	A	A	H	L		L	H	A	A	A	A											
7			39	A	A	A	A	L	H	GND	H	L	A	A	A	B	4.5V			All Outputs	H or L as shown						VDC

Notes:
1. L = 0.4V max. H = 2.4V min.
2. A = 2.0V min. B = 0.7V max.
3. Test must be run in sequence as shown.

TEST NO. 5.9

Logic Integrity Test for the 54LS174 (Vendor C) Hex D-type Flip-Flop

Objective:

Develop a Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 06 (54LS174 Hex D-Type Flip-Flops).

Circuit Description:

The 54LS174 contains six positive edge-triggered D-Type flip-flops implemented with Schottky TTL circuitry. The six flip-flops have a common clock input and a common direct clear input. Information at the D inputs meeting set-up time requirements is transferred to the Q outputs on the positive going edge of the clock pulse. Clock triggering occurs at a particular voltage level and is not directly related to the transition time of the positive going pulse. When the clock input is either high or low, the D input has no effect at the output.

Test Plan:

I. Review the Schematic Diagram (Figure 1C) presented in MIL-M-38510/301 to determine if the circuit mechanization presented is functionally correct.

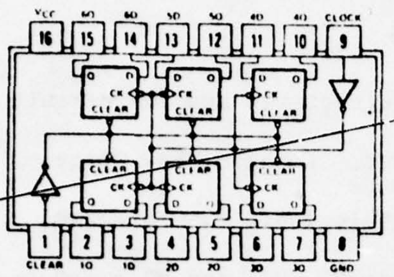
   a. Develop a Logic Flow Diagram from the Schematic supplied.

   b. Determine if the Logic Flow Diagram of the 54LS174 is identical to the 54S174.

      1. If identical, use the 54S174 vectors.

      2. If different, generate a new Logic Integrity Test.

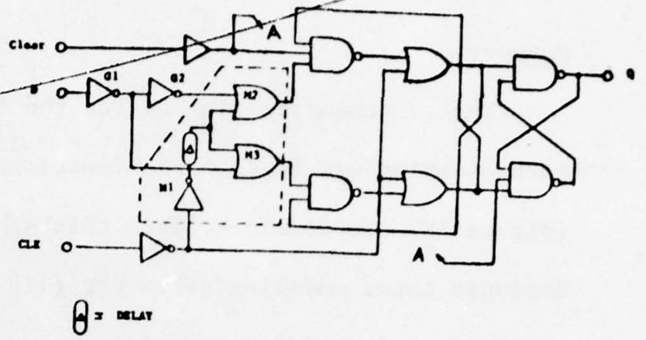
II. Test a device with the vectors.



Cases F and F

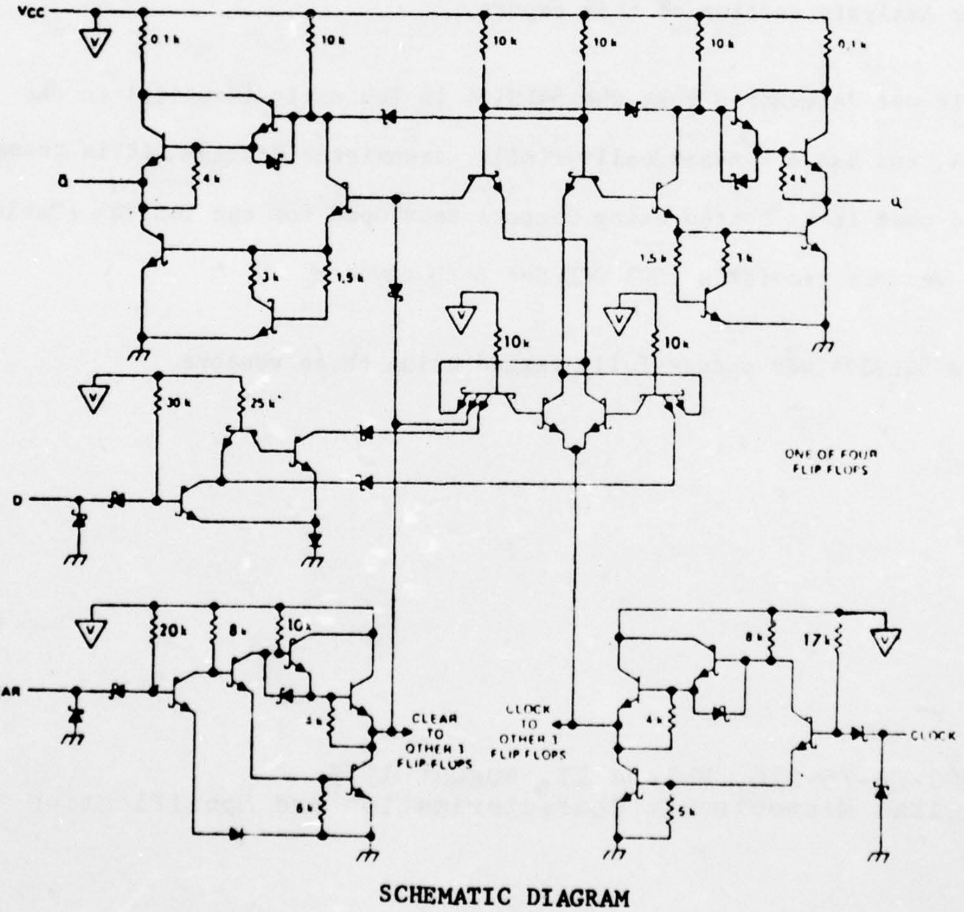
BLOCK DIAGRAM

(A)



LOGIC DIAGRAM

(B)



SCHEMATIC DIAGRAM
(C)

FIGURE 1

Summary:

The Schematic Diagram for the 54LS174 was reviewed and the circuit mechanization was found to be functionally correct. A Logic Flow Diagram (Figure 1B) was developed from this Schematic. This logic flow diagram includes three modeling gates per flip flop (gates M1, M2, and M3 - not realized as individual transistors in the hardware) which were introduced to simulate the operation of transistors Q_A, Q_B, Q_C, and Q_D, (Figure 2B) during a positive transition of the clock input.

The modeling gates and the operation which they simulate are discussed in the Analysis section of this report.

It was determined that the 54LS174 is logically identical to the 54S174, and has a fundamentally similar transistor diagram. It is recommended that it be tested using vectors developed for the 54S174* (Table 1). These vectors provide a 100% TCL for both devices.

A 54LS174 was successfully tested using these vectors.

*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

Analysis

Transfer of data from the D input to the latch is accomplished by relative voltage levels associated with transistors Q_C and Q_D to achieve a transition sensitive clock function. Figure 2A shows the circuitry used to implement these gates. Due to the voltage drop across diodes D1 and D2 the output of this circuit cannot drop below ≈1.7 volts (2 X .7 + VCE (sat)). Thus even when the input is high the output represents a logic "1" except during a positive clock transition. This is due to the following circuit operation (see Figure 2B):

> When the clock goes from low to high transistor Q1 conducts, activating transistors Q_A and Q_B. The bistable latch formed by transistors Q_A, Q_B, Q_C, and Q_D will assume one of two states depending on the voltage present at nodes A and B (outputs of G1 and G2). One of these nodes will be at a significantly higher voltage than the other (VCC compared with 1.7V). This will cause a larger current to be diverted into the base of its associated OR gate transistor (Q_A or Q_B). The result is that either Q_A or Q_B reaches saturation ahead of other and turns the other off through feedback line F1 or F2.

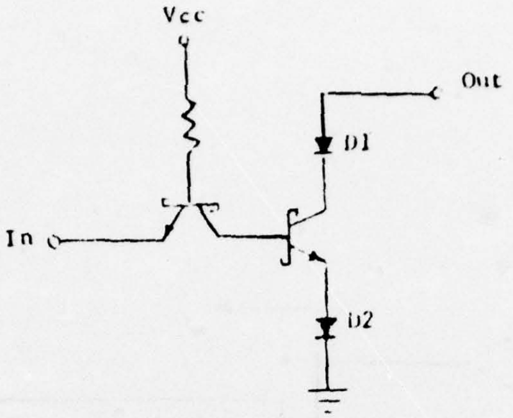
The effect of this circuit operation is to make a voltage of 1.7V, at either node A or B, appear as a temporary logic "O" during a positive clock transition while the latch is accepting input data. In all other cases, however, it represents a logic "1", thereby isolating the D input.

It should be noted that there are two outputs of G1; one which drives G2 is taken from the collector of Q_E and always appears as $\overline{D}$ to G2, and one which drives Q_D through D1 which can only appear as logic "0" during the clock transition.

The operation described above was modeled as shown in Figure 3. Inverter M1 was given sufficient propagation delay to allow a logic "0" from G1 or G2 to be recognized by the latch before the rising clock forces the outputs of M2 and M3 to logic "1", thus simulating the actual circuit operation.

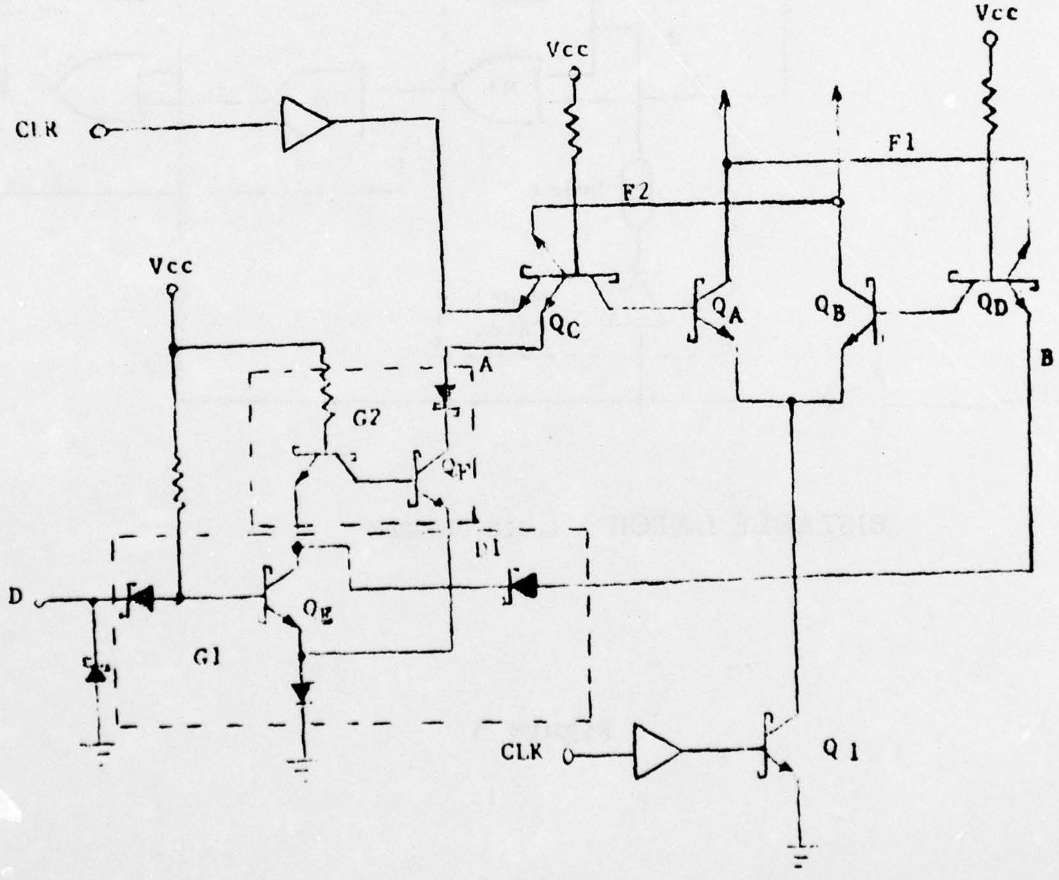
Due to the mechanization of inverters G1 and G2, the output (node A or B) S-A-0 is not the same as the input S-A-1, therefore, an additional test is required on each inverter. These tests are included in the vectors presented in Table 1.

There are five possible faults which lead to an indeterminate state. These are S-A-1 faults on inputs to G3 and G4 fed by G1 and G2, S-A-1 faults on the outputs of G1 and G2, and a S-A-1 fault on the input of G2. If any of these faults exists, and the timing is such that the function is affected, the fault will be caught by the vectors presented in Table 1. If the internal gate delays are such that the fault does not affect the function, the fault is considered irrelevant. With this qualification, a TCL of 100% is achieved with the vectors presented in Table 1.



TRANSITION SENSITIVE CIRCUIT

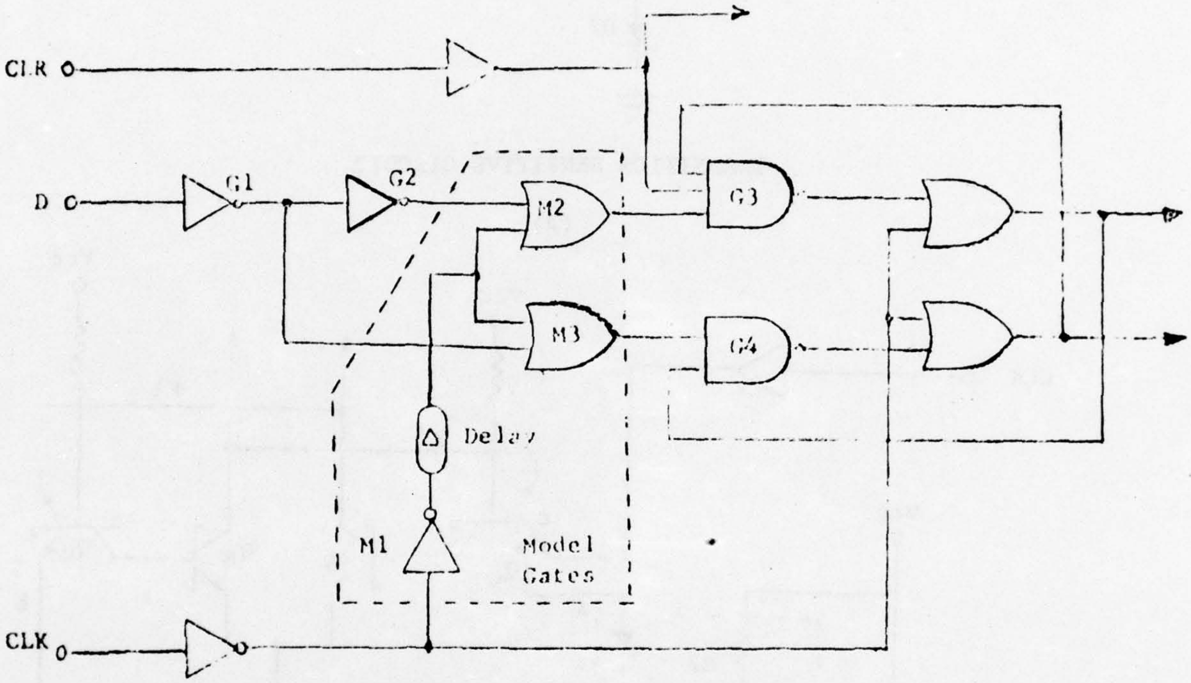
(A)



BISTABLE LATCH - SCHEMATIC DIAGRAM

(B)

Figure 2



BISTABLE LATCH - LOGIC DIAGRAM

Figure 3

TABLE 1

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																MEASUREMENT TERMINAL	TEST LIMITS						UNITS
			E	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	CLR	1Q	1D	2D	2Q	3D	3Q	GND	CLK	4Q	4D	5Q	5D	6D	6Q	Vcc		Min	Max	Min	Max	Min	Max	
7			1	B	L	A	A	L	A	L	GND	A	L	A	L	A	A	L	4.5V	ALL Outputs	H or	L as shown					VDC
			2	B	L	A	A	L	A	L		B	L	A	L	A	A	L									
			3	B	L	A	A	L	A	L		A	L	A	L	A	A	L									
			4	A	L	A	A	L	A	L		A	L	A	L	A	A	L									
			5	A	L	A	A	L	A	L		B	L	A	L	A	A	L									
			6	A	H	A	A	H	A	H		A	H	A	H	A	A	H									
			7	A	H	B	B	H	B	H		A	H	B	H	B	B	H									
			8	A	H	B	B	H	B	H		B	H	B	H	B	B	H									
			9	A	L	B	B	L	B	L		A	L	B	L	B	B	L									
			10	A	L	A	A	L	A	L		B	L	A	L	A	A	L									
			11	A	H	A	A	H	A	H		A	H	A	H	A	A	H									
			12	A	H	A	A	H	A	H		B	H	A	H	A	A	H									
7			13	B	L	A	A	L	A	L	GND	B	L	A	L	A	A	L	4.5V	ALL Outputs	H or	L as shown					VDC

NOTES:

1. L = 0.4V max. H = 2.4V min.
2. A = 2.0V min. B = 0.7V max.
3. Test must be run in sequence shown.

TEST NO. 5.10

Logic Integrity Test for the
54LS175 (Vendor C) Quad D-type Flip-Flop

.Objective:

Develop a Logic Integrity Test (LIT) for MIL-M-38510/301 Device Type 07 (54LS175 Quad D-Type Flip-Flops).

Circuit Description:

The 54LS175 contains four positive edge-triggered D-Type flip-flops implemented with Schottky TTL circuitry. The four flip-flops have a common clock input and a common direct clear input. Information at the D inputs meeting set-up time requirements is transferred to the outputs on the positive going edge of the clock pulse. Clock triggering occurs at a particular voltage level and is not directly related to the transition time of the positive going pulse. When the clock input is either high or low, the D input has no effect at the output.

Test Plan:

I. Review the Schematic diagram (Figure 1C) presented in MIL-M-38510/301 to determine if the circuit mechanization presented is functionally correct.

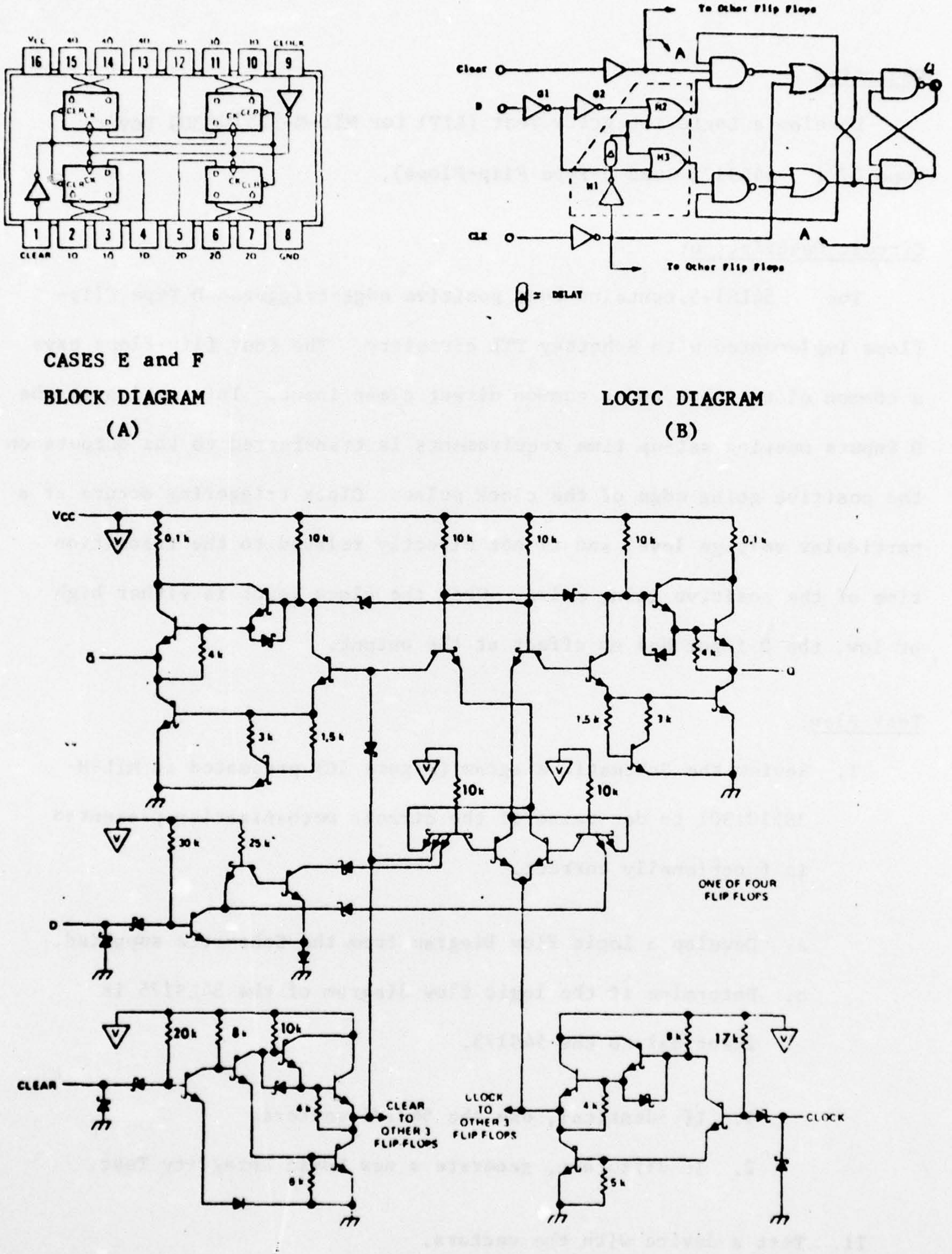
   a. Develop a Logic Flow Diagram from the Schematic supplied.

   b. Determine if the logic flow diagram of the 54LS175 is identical to the 54S175.

      1. If identical, use the 54S175 vectors.

      2. If different, generate a new Logic Integrity Test.

II. Test a device with the vectors.



CASES E and F

BLOCK DIAGRAM

(A)

LOGIC DIAGRAM

(B)

SCHEMATIC DIAGRAM

(C)

FIGURE 1

Summary:

The schematic diagram for the 54LS175 was reviewed and the circuit mechanization was found to be functionally correct. A Logic Flow Diagram (Figure 1B) was developed from this Schematic. This Logic Flow Diagram includes three modeling gates per flip flop (gates M1, M2, and M3 - not realized as individual transistors in the hardware) which were introduced to simulate the operation of transistors Q_A, Q_B, Q_C, and Q_D, (Figure 2B) during a positive transition of the clock input.

The modeling gates and the operation which they simulate are discussed in the Analysis section of this report.

It was determined that the 54LS175 is logically identical to the 54S175 and has a fundamentally similar transistor diagram. It is recommended that it be tested using vectors developed for the 54S175* (Table 1). These vectors provide 100% TCL for both devices.

A 54LS175 was successfully tested using these vectors.

*RADC-TR-75-216, Volume II, August 1975
Digital Microcircuit Characterization and Specification

Analysis

Transfer of data from the D input to the latch is accomplished by relative voltage levels associated with transistors Q_C and Q_D to achieve a transition sensitive clock function. Figure 2A shows the circuitry used to implement these gates. Due to the voltage drop across diodes D1 and D2 the output of this circuit cannot drop below 1.7 volts (2 X .7 + VCE (sat)). Thus even when the input is high the output represents a logic "1" except during a positive clock transition. This is due to the following circuit operation (see Figure 2B):

> When the clock goes from low to high transistor Q1 conducts, activating transistors Q_A and Q_B. The bistable latch formed by transistors Q_A, Q_B, Q_C, and Q_D will assume one of two states depending on the voltage present at nodes A and B (outputs of G1 and G2). One of these nodes will be at a significantly higher voltage than the other (VCC compared with 1.7V). This will cause a larger current to be diverted into the base of its associated OR gate transistor (Q_A or Q_B). The result is that either Q_A or Q_B reaches saturation ahead of other and turns the other off through feedback line F1 or F2.

The effect of this circuit operation is to make a voltage of 1.7V, at either node A or B, appear as a temporary logic "0" during a positive clock transition while the latch is accepting input data. In all other cases, however, it represents a logic "1", thereby isolating the D input.

It should be noted that there are two outputs of G1; one which drives G2 is taken from the collector of Q_E and always appears as $\overline{D}$ to G2, and one which drives Q_D through D1 which can only appear as logic "0" during the clock transition.

The operation described above was modeled as shown in Figure 3. Inverter M1 was given sufficient propagation delay to allow a logic "0" from G1 or G2 to be recognized by the latch before the rising clock forces the outputs of M2 and M3 to logic "1", thus simulating the actual circuit operation.

Due to the mechanization of inverters G1 and G2, the output (node A or B) S-A-0 is not the same as the input S-A-1, therefore, an additional test is required on each inverter. These tests are included in the vectors presented in Table 1.

There are five possible faults which lead to an indeterminate state. These are S-A-1 faults on inputs to G3 and G4 fed by G1 and G2, S-A-1 faults on the outputs of G1 and G2, and a S-A-1 fault on the input of G2. If any of these faults exists, and the timing is such that the function is affected, the fault will be caught by the vectors presented in Table 1. If the internal gate delays are such that the fault does not affect the function, the fault is considered irrelevant. With this qualification, a TCL of 100% is achieved with the vectors presented in Table 1.

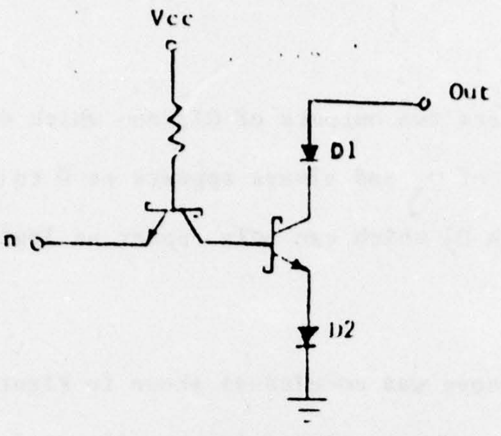
AD-A038 969   GENERAL ELECTRIC CO PITTSFIELD MASS ORDNANCF SYSTEMS   F/G 9/5
DIGITAL MICROCIRCUIT CHARACTERIZATION AND SPECIFICATION.(U)
MAR 77   T M OSTROWSKI   F30602-74-C-0159

UNCLASSIFIED   RADC-TR-77-91   NL

3 OF 3
AD
A038969

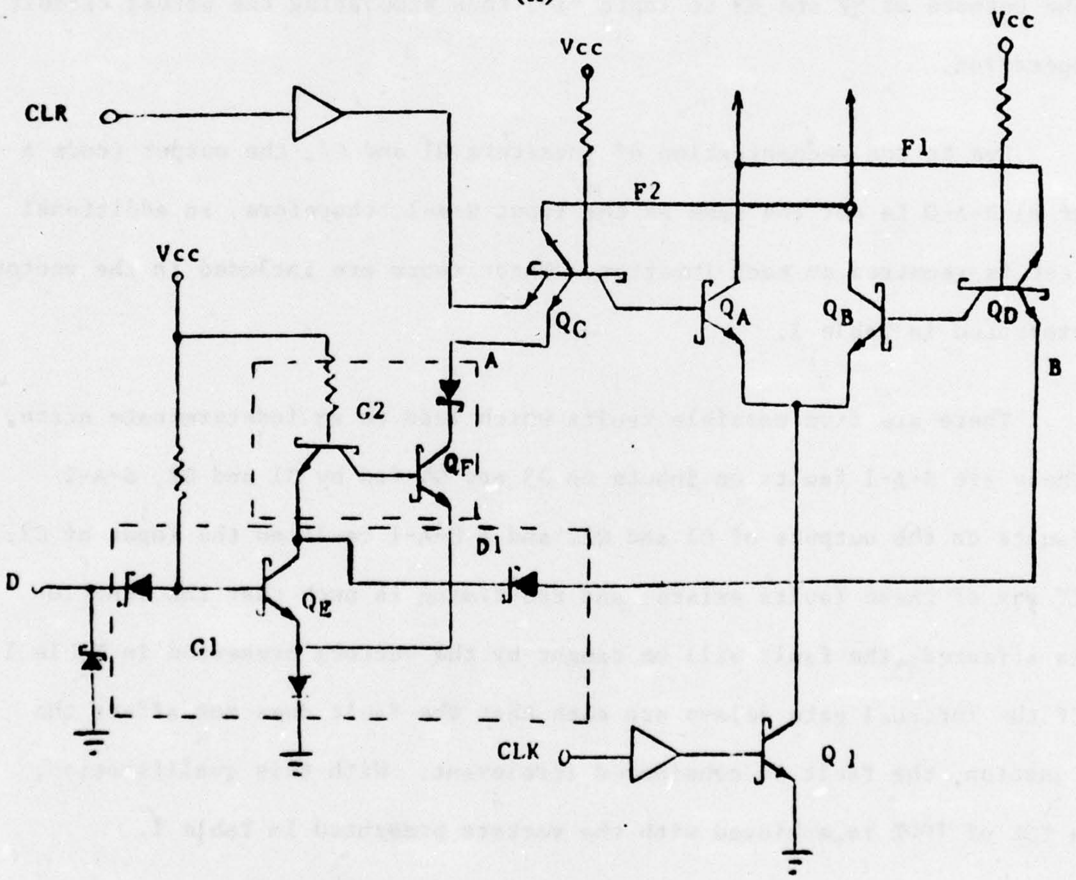
END

DATE
FILMED
5-77



TRANSITION SENSITIVE CIRCUIT

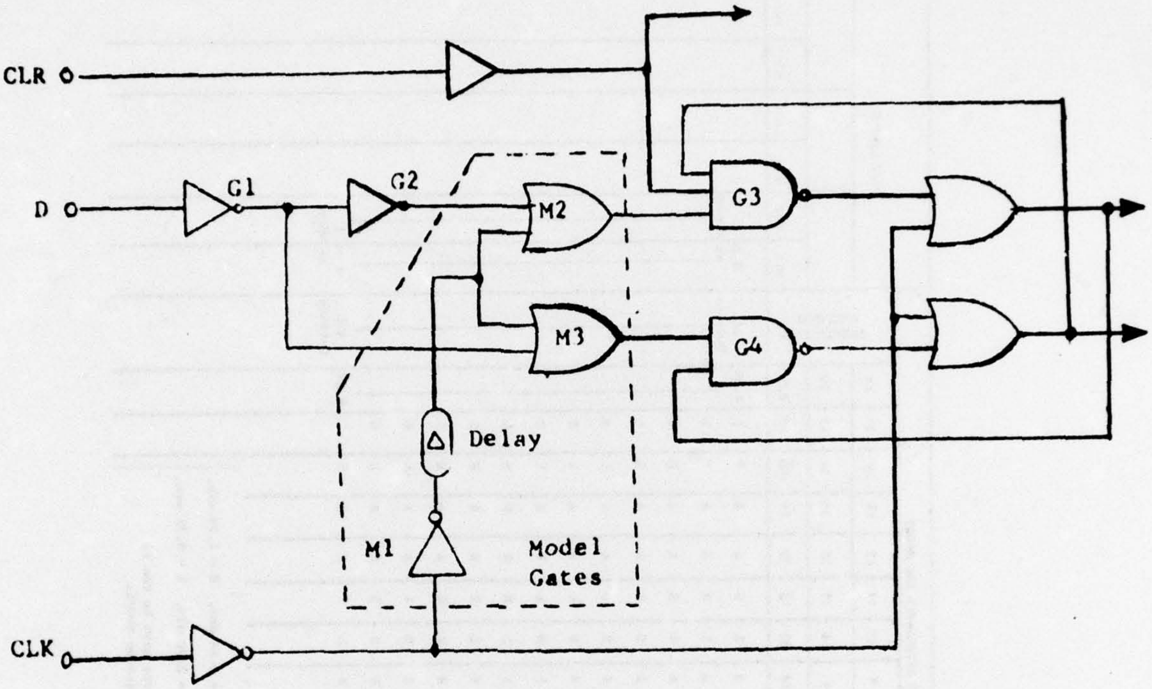
(A)



BISTABLE LATCH - SCHEMATIC DIAGRAM

(B)

Figure 2



BISTABLE LATCH - LOGIC DIAGRAM

Figure 3

SUBGROUP	SYMBOL	MIL-STD-883 METHOD	CASE	TERMINAL CONDITIONS (PINS NOT DESIGNATED ARE OPEN)																	TEST LIMITS						UNITS
			D-2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	MEASUREMENT TERMINAL							
			F-5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
			TEST NO.	CLR	1Q	1$\overline{Q}$	1D	2D	2$\overline{Q}$	2Q	GND	CK	3Q	3$\overline{Q}$	3D	4D	4$\overline{Q}$	4Q	VCC		Min	Max	Min	Max	Min	Max	
7			1	B	L	H	A	A	H	L	GND	B	L	H	A	A	H	L	4.5V	ALL Outputs	H or	L as shown					VDC
			2	B	L	H	A	A	H	L		A	L	H	A	A	H	L									
			3	B	L	H	A	A	H	L		B	L	H	A	A	H	L									
			4	A	L	H	A	A	H	L		B	L	H	A	A	H	L									
			5	A	H	L	A	A	L	H		A	H	L	A	A	L	H									
			6	A	H	L	A	A	L	H		B	H	L	A	A	L	H									
			7	A	H	L	B	B	L	H		B	H	L	B	B	L	H									
			8	A	L	H	B	B	H	L		A	L	H	B	B	H	L									
			9	A	L	H	B	B	H	L		B	L	H	B	B	H	L									
			10	A	L	H	A	A	H	L		B	L	H	A	A	H	L									
			11	A	H	L	A	A	L	H		A	H	L	A	A	L	H									
			12	A	H	L	A	A	L	H		B	H	L	A	A	L	H									
7			13	B	L	H	A	A	H	L	GND	B	L	H	A	A	H	L	4.5V	ALL Outputs	H or	L as shown					VDC

NOTES:
1. L = 0.4V max. H = 2.4V min.
2. A = 2.0V min. B = 0.7V max.
3. Tests must be run in sequence shown.

APPENDIX

Table of Contents

APPENDIX A

APPENDIX B

APPENDIX

List of Figures

APPENDIX

List of Tables

APPENDIX

List of Tables, Cont'd

APPENDIX A

SECTION A.1

EVALUATION OF A FUNCTIONAL TEST SET FOR THE Mc6800 MICROPROCESSOR

INTRODUCTION

Vendor A will supply a set of functional tests (1376 vectors) with the purchase of an Mc6800. This test set (Functional Test Set I) was developed on a FAIRCHILD SENTRY 500 Tester which is limited to 1024 vectors per buffer. therefore, the vectors are supplied in two parts (The long and short functional tests). Both parts are required for a complete functional test.

Due to the complexity of this and other VLSI (Very Large Scale Integration) devices, the development and use of all possible input vectors for each state of the machine would be impractical. Therefore, for either test generation or test evaluation, a model representing the internal functional blocks of the uP (microprocessor) must be used which identifies the internal machine states and interconnecting data paths. This should include a gate level logic diagram, timing diagram and a block diagram which shows the major functional areas of the uP and the interconnecting data and control paths. Since neither a logic diagram nor a sufficiently detailed block diagram was available, the approach taken to evaluate the effectiveness of the functional test was to:

1) Section the uP into functional blocks and develop a block diagram based on available information.
2) Contact Vendor A personnel regarding internal operation of the Mc6800 and the correctness of this diagram.
3) Analyze the test vectors to determine what rationale was used and the validity and completeness of this rationale with respect to verification of the blocks and instruction set.
4) Discuss the evaluation with Vendor A personnel.
5) Based on experience, evaluate the effectiveness of the vectors in comparison with the type of tests required for each functional block. In order to evaluate the test vectors more effectively and efficiently, a computer program was written to convert the binary representation of the vectors to an assembler language level of representation. This program will be referred to in this report as the "disassembler".

SUMMARY

In order to evaluate these test vectors the Mc6800 function was sectored into functional blocks with observable states. After an initial review of the Mc6800 block diagram and instruction set, it was determined that the function could be sectored into three broad categories:

1) the Machine Control area which includes function areas such as timing, interrupt control, instruction decode, branching logic and peripheral interfacing;
2) the Arithmetic sections including the ALU, one's complementor, shifter and condition deteotors (zero, negative etc. for arithmetic and logic operation);
3) the Registers and associated Increment/Decrement Circuitry.

These sectors were further subdivided as necessary to evaluate the vectors.

The results of the evaluation using the resultant block diagram is as follows:

The operation of the data and output buffers, data and address buses and timing and control section was tested by the vector sets, and no additional vectors are required. The Instruction Decode, Arithmetic and Logic Unit and Register Array sections require additional vectors to achieve the desired degree of testing confidence.

During this evaluation, Vendor A supplied a new set of test vectors (Test Set II). This set was developed for two reasons:

1) to include failure analysis data obtained since the first sets were developed.
2) to shorten the test by removing redundant tests so that the complete test set would fit in one SENTRY 500 data buffer. Note: the extra vectors were originally included for fault isolation during design evaluation but aren't necessary for fault detection.

For these reasons Vendor A is now using the new vector set for production testing of Mc6800's. Because recent failure analysis data is included in Test Set II, Vendor A recommends that this set be used.

The detailed analysis of the following areas for the long and short functional tests of Test Set I was not completed.

1) Determination that all input combinations applied to the ALU were sensitized to the output.
2) Determination that the correct operation of all op codes were verified at the outputs.

These parts of the analysis were not completed since the new test set had been received and it would be more advantageous to expend further effort evaluating the new test set.

It was observed during this evaluation of Test Set I that:

1) Two op codes were used which are not specified in the data sheets for the Mc6800. These are 00 and 03. These are NOP's and appear to be BITE (Built In Test Electronics).
2) The Conditional Branches were used extensively, some as many as 25 times. These were included for fault isolation during design verification of the uP. They are not all needed for fault detection.
3) Several operations on a particular register were frequently concatenated without analyzing the intermediate results. This method of testing can cause the results of a fault to be masked on succeeding vectors such that fault is not detected.
4) The instruction decode section was mechanized using two decode sectors. Fewer test vectors are required to test this mechanization.
5) In discussions with Vendor A, they stated that these vectors were developed as a design verification tool. Vectors were added as required during the design evaluation. As a result the test vectors do not follow a particular plan.
6) Vendor A used knowledge of the device in the development of their test sets. Without a logic diagram, it was not possible to verify the effectiveness of some of their vectors such as the NOP.

DISCUSSION

Backgrounꞁ

The block diagram (Figure A-1) contained in Vendor A's data sheet shows the basic architecture of the Mc6800. The instruction decode and control section and the input/output parts are functional blocks which are required of the central processor in any computer. In a fixed bit length uP such as the Mc6800 (as opposed to a bit slice uP such as the Vendor C SBP0400) these sections are the boundaries of the uP chip, that is, the sections which interface with the rest of a system, and specify the operating mode of each internal section of the uP. The actual operation of these sections will be explained later in the report. The primary internal sections of the uP are the ALU (Arithmetic and Logic Unit), registers, internal data buses and instruction decode and control section. A person wishing to use this chip is initially interested in the following:

1) the width of the buses (number of bits),
2) the quantity and quality of operations that the ALU and associated circuitry can perform,
3) the quantity and size of the registers, and
4) the operations that can be performed on each register.

Much of this information can be obtained from the block diagram (Figure A-1), a list of op codes and their functions and the brief description contained in the Vendor A data sheet. However, for the purposes of test generation and test evaluation, this block diagram is lacking in much of the required detail. For example, the instruction decode and control is shown as a box with 16 inputs and three outputs. In actuality, this block has additional inputs from the Condition Code Register and probably hundreds of outputs controlling the registers, I/O ports, ALU and data buses. In order to evaluate the test vector set, an understanding is required of the function of this box and its interaction with the other sections. To aid in this understanding, expanded block diagrams were developed for this and other sections.

The supplied test vectors were presented as a table of "ones" and "zeros". If a logic diagram were made available, computer simulation programs could be used to evaluate their effectiveness. Since a logic diagram was not available, the test vector set effectiveness was determined by analyzing the functional requirements of each functional block of the 6800.

Since this method requires a description of the activity of each vector, a computer program, the "disassembler", was developed to convert the table of vectors to an assembly language equivalent.

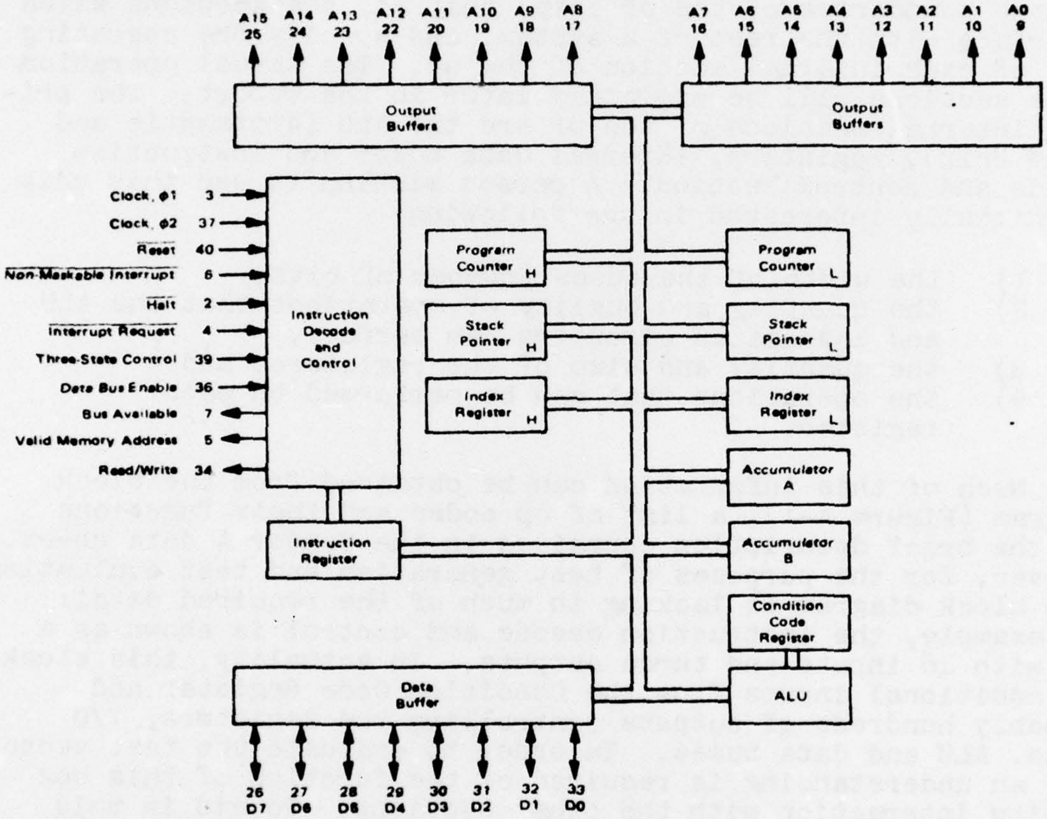


Figure A-1. Mc6800 Block Diagram

Vector 17 was the first executable op code. The first 16 vectors were used to initialize the uP. Therefore, Vector 17 was used as a starting point for the disassembler. The disassembler decoded the data lines as op code B6 and the address as 0C63. This op code was identified as a LDAA (Load Accumulator A from the memory location pointed to by the next two vectors which are called Byte 2 and Byte 3). This instruction was also identified as a four-byte (four-vector) instruction. Therefore, the next op code appears at 17 plus 4 or Vector 21 (a LDAB instruction). This procedure is continued until the entire test set is disassembled.

If the uP were to read this test set from RAM, it would not be executed correctly. This is due to the fact that the program instructs the uP to store data into the memory locations where other instructions would be located. However, these vectors were never intended to be a program executed from RAM. They are instead an ordered set of vectors applied by an automatic tester.

When these op codes were analyzed, it was found that the contents of the registers were often modified many times before the data was made visible at the external chip pins. In order to evaluate the effectiveness of many of the instructions, the data in the registers had to be determined for each vector. An interpreter program was written by Vendor A which can provide this information and is available on several of the nationwide dial-up time-sharing facilties. This program reads a file as though it were a RAM loaded with a program and prints out various actions which would be performed by a uP. This program was used to determine the register contents for the long test. In order to use the Interpreter, the test vectors had to be modified so that the program would not write on top of itself. The result is that the program counter is not always correct. The resulting listing was modified so that the program counter would be represented accurately.

These computer printouts and expanded block diagrams were used to evaluate the various functional blocks.

Data and Output Buffers

The data and output buffer lines along with the read/write line interface the Mc6800 with memory (RAM and ROM) and input/output devices (teletype, line printer, floppy disc, modem, etc.). The address lines select a particular I/O device or a particular memory address. If the read/write line is forced high by the uP then the selected memory or input device will place data from the selected location onto the data bus. Once the data is stable (determined when the Phase 2 clock is brought high) the 6800 reads this data and puts it in the particular register specified by the instruction that it is executing. If the data is loaded into the Instruction Register then it is used as the next instruction. When the read/write line is brought low by the uP and RAM is addressed, then the uP will put data on the data bus. When the data is stable, it will be written into the selected location of RAM. Likewise, if an output device is specified such as the PIA in Figure A-2 (Peripheral Interface Adapter), then the data on the data bus is output to that device (e.g., to a printer or tape).

The circuitry analyzed in this section includes:

1) The data line buffers and associated chip pins,
2) An 8-bit data bus connecting data buffers, the registers, and the ALU,
3) The address buffers and associated chip pins,
4) A 16-bit address bus from the program counter, stack pointer, index register and address incrementor/decrementor to the address buffers.

Each flip-flop in the data and address buffers was checked for 1 to 1, 1 to 0, 0 to 0, and 0 to 1 transitions. In addition, each data bus line and therefore, the associated buffer flip-flops and chip pins, were shown to be independent from each other.

Operation of the data and output buffers was verified. No additional test vectors are required.

Instruction Decode and Control

The first sector broken off as a separate entity from the Instruction Decode and Control Block was the basic timing and control section (Figure A-3a). The inputs to this section, for example, interrupts, halt, etc., determine operation of the uP independent of the op codes which are being executed.

The clock inputs are inherent to all aspects of the microprocessor's operation. The clock lines are automatically verified if the vector set is sufficent for each functional block. E.g., one of the clocks defines the time when a particular register is loaded with data. If the line is faulted, the register will either not be loaded or will be loaded even when loading is not specified. Both of these cases will be detected as bad data in the register.

The interrupt control lines ($\overline{\text{RESET}}$, $\overline{\text{HALT}}$, $\overline{\text{NONMASKABLE INTERRUPT}}$ ($\overline{\text{NMI}}$), and $\overline{\text{INTERRUPT REQUEST}}$ ($\overline{\text{IRQ}}$)), are used in situations where a low priority activity must be interrupted so that a higher priority activity can be executed. After the high priority task is completed, the low priority task is resumed.

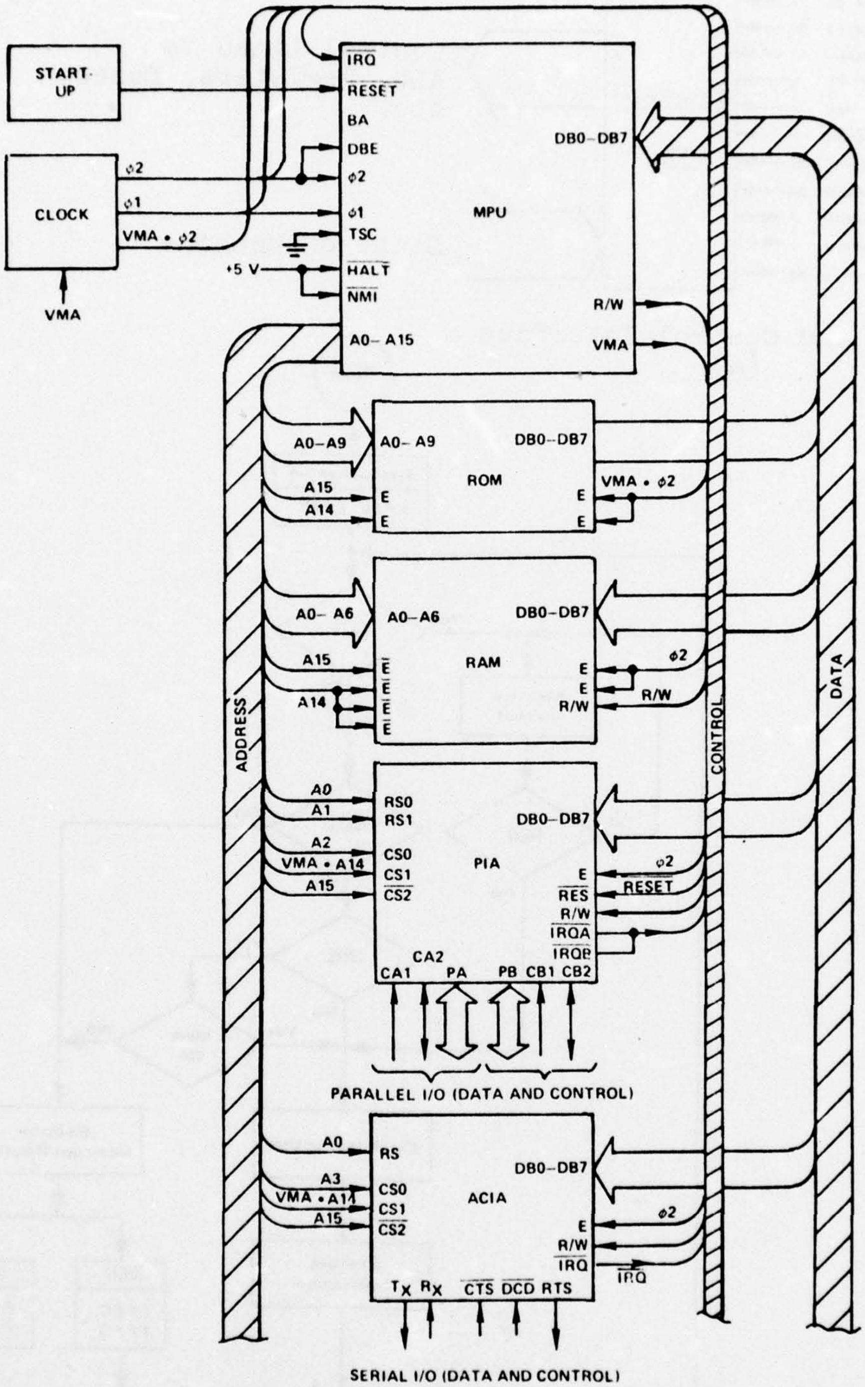
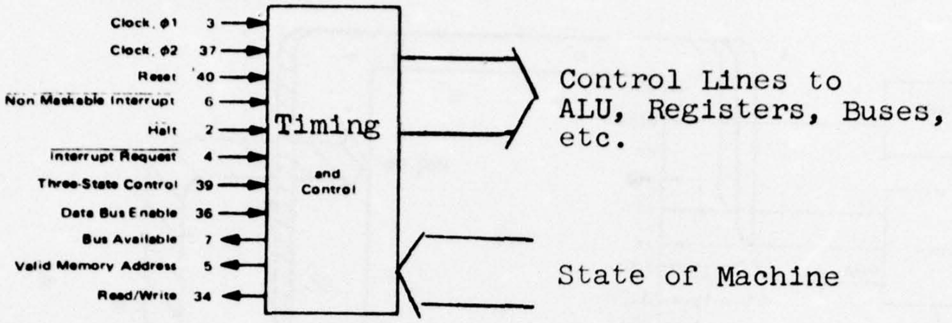
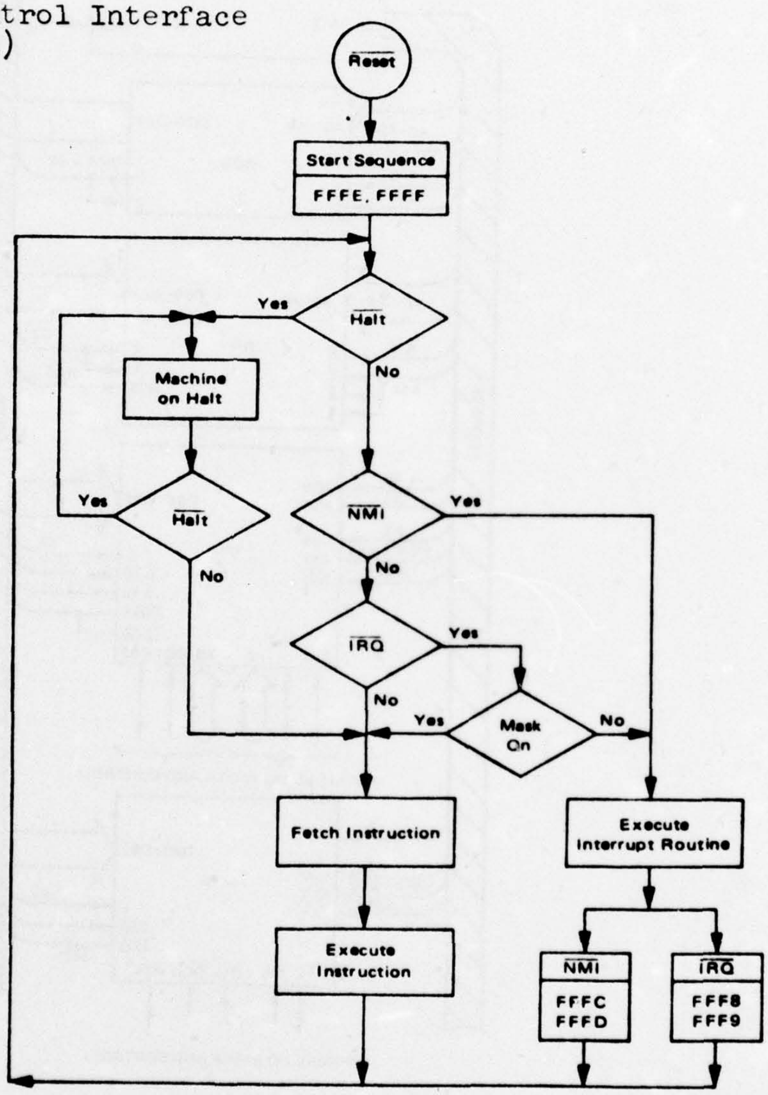


Figure A-2. Typical Mc6800 System Wiring


Clock, φ1 3
Clock, φ2 37
$\overline{\text{Reset}}$ 40
$\overline{\text{Non Maskable Interrupt}}$ 6
$\overline{\text{Halt}}$ 2
$\overline{\text{Interrupt Request}}$ 4
Three-State Control 39
Data Bus Enable 36
Bus Available 7
Valid Memory Address 5
Read/Write 34
Timing
and Control
Control Lines to ALU, Registers, Buses, etc.
State of Machine

Timing and Control Interface
(a)



(b) Interrupt Management of Mc6800

Figure A-3. Control Sector of the Mc 6800.

E.g., a uP could search through and modify a list contained in RAM concurrently with accepting data from a teletype and store it elsewhere in RAM. The search and modify task is a time-consuming task but one which can be interrupted and resumed without any loss of accuracy. The teletype task, on the other hand, is not time-consuming nor is it frequent relative to the uP's execution speed, but the task must be executed during the short period of time when the data is available. This task is easily handled using the interrupt capability. After each instruction of the table search is executed, the uP will check for a low on $\overline{\text{HALT}}$, $\overline{\text{NMI}}$ or $\overline{\text{IRQ}}$, and if any of these conditions are true, then the action indicated in Figure A-3b is performed. By wiring the teletype to force NMI low when it is ready with data, the full power of the uP can be used on the table search until an interrupt is received by the uP. Then, the uP

1) stores the registers into the area of RAM which is addressed by the Stack Pointer Register,
2) loads the Program Counter with the contents of memory locations hex (hexadecimal) FFFC and FFFD, which contain the user-defined starting location of the teletype servicing routine,
3) executes the teletype routine,
4) reloads the registers from the stack, and
5) continues with the search routine. Note that the Program Counter is stored into the stack at the beginning of the interrupt and restored at the end. Since the Program Counter always contains the address of the next instruction to be executed, the uP can resume the low priority task exactly where it left off.

The other interrupts and control are:

1) $\overline{\text{IRQ}}$ which, if the Mask bit in the Conditions Code Register is zero, affects the uP as described for $\overline{\text{NMI}}$ except that the Program Counter is loaded from memory locations hex FFF8 and FFF9.
2) $\overline{\text{RESET}}$ which causes the contents of memory locations hex FFFE and FFFF to be loaded into the Program Counter.
3) SWI (Soft ware Interrupt), a software instruction, is similar to $\overline{\text{NMI}}$ except that locations hex FFFA and FFFB are used to load the program counter.
4) $\overline{\text{HALT}}$ which is not an interrupt in the sense of the other four but has a higher priority than $\overline{\text{NMI}}$, $\overline{\text{IRQ}}$, and SWI. Its function is to suspend operations of the uP.

A test of these functions should check that:

1) each performs the intended function.
2) The proper priority is maintained when any two or more are detected at the same time.
3) Each will perform independent of the previous instruction.

In the test set each of these functions is exercised. Several combinations of these lines are activated together which verified the hierarchy of the priorities as shown in Figure A-3b. Figure A-3b indicates that these checks are independent of the instruction being executed when an interrupt or halt occurs. A logic diagram is required for verification. Based on available information a satisfactory test is accomplished on this area.

The remaining external leads are Three-State Control, Data Bus Enable, Bus Available and Valid Memory Address. These lines and their associated circuitry are exercised and their operation is verified provided that the three-state lines are checked for the high impedance state. This should be verified by a parametric test. The internal control and sense lines shown in Figure A-3a interface the Instruction Decode and Control Section with the other sectors of the uP. It is best to evaluate these lines on an individual basis as part of the sector with which it interfaces. As such they will not be mentioned specifically nor considered a part of timing and control for evaluation but rather implicit in the operation of each sector as it is evaluated.

In summary, the test set verifies the function of the Timing and Control Sector.

The second sector of the Instruction Decode and Control section is the Instruction Decode sector. In order to describe what is involved in testing this section of a uP the operation and a possible implementation of this will first be described.

A program instruction is initiated when the microprocessor addresses a location in memory and loads the data into its Instruction Register. This data is known as an op code and defines the function that the uP is to perform (e.g. ADD, BRANCH, etc.). Once the op code has been loaded into Instruction Register, control circuitry is activated in the instruction decode area which is unique to that instruction. In order for this circuitry to be tested, data must be available on the input buses and internal registers which will distinguish that the particular instruction was excuted correctly and that no other control circuitry (wrong instruction) was activated instead. If the result is contained in a register, then another instruction has to be executed which will bring the information to an externally observable point. Referring to Figure A-1, the op code is brought in through the data buffer onto the data bus and then into the Instruction Register during the first clock cycle of each instruction. This eight-bit code is applied to the inputs of a decoder tree (Figure A-4a), which in turn activates (e.g., pulls low) one of the outputs (instruction lines) of the decoder provided that the op code is a valid instruction. This instruction line activates a number of control lines which go directly to atomic functional areas of the uP. This is done by pulling the control line to the same logic level by means of diodes, shown as dots in the Programmed Logic Array (PLA) section of Figure A-2a. For example, consider the ADA instruction which adds

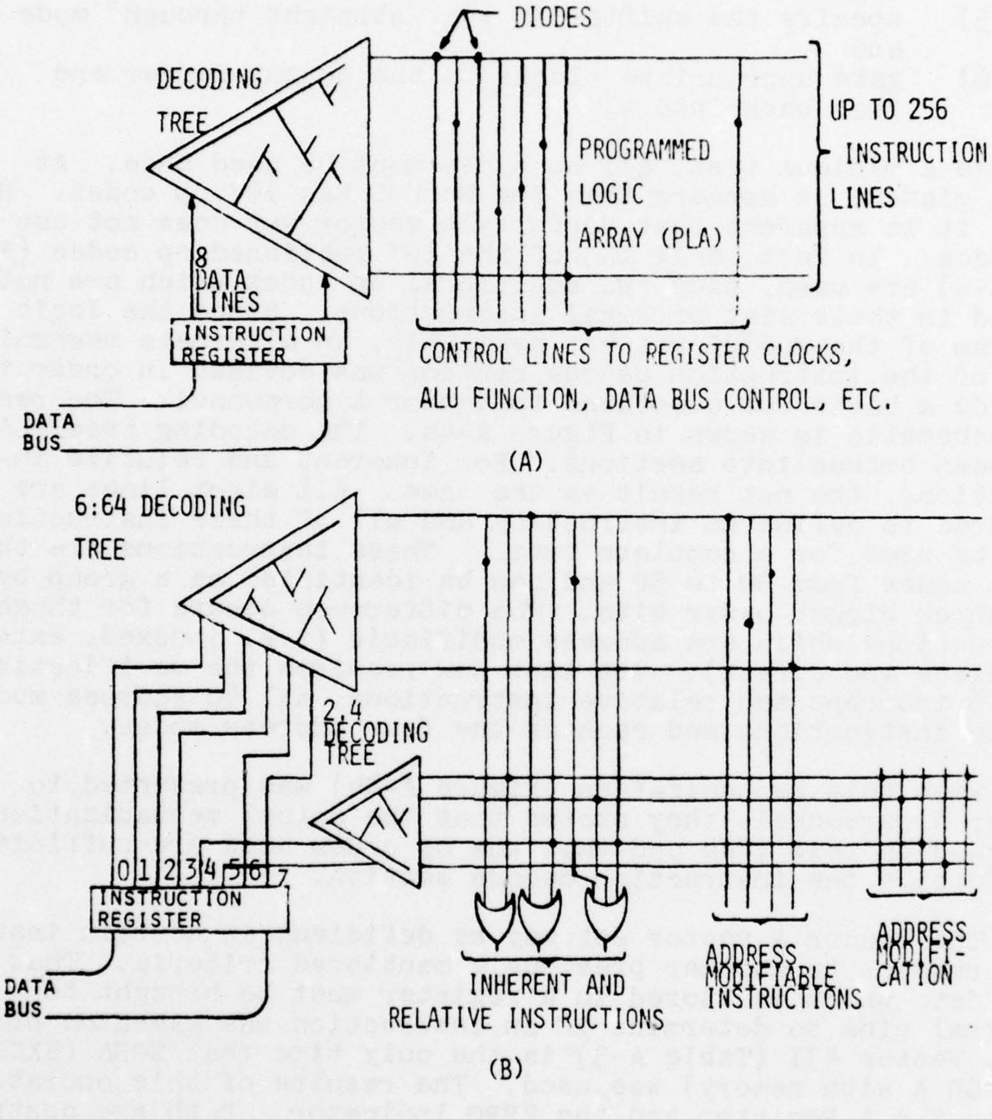


Figure A-4. Microprocessor Instruction Decode Circuit

Accumulator A to Accumulator B and places the result in Accumulator A. The instruction line must activate the gates which:

1) connect the accumulators through the ALU.
2) place the true/complement circuit in the true mode,
3) set the ALU in the "ADD" mode,
4) gate the appropriate bits of the Condition Code Register,
5) specify the shifter in the "straight through" mode and
6) gate appropriate clocks to the output buffer and then back into A.

As a minimum test, all op codes must be used once. At first glance, it appears that the Mc6800 has 197 op codes. However, it is apparent that Vendor A's vector set does not use all op codes. In fact, only 123 of the 197 published op codes (Figure A-4) are used, plus two additional op codes which are not listed in their list of legal instructions. Since the logic diagram of the Mc6800 was not available, an alternate mechanization of the instruction decode section was devised in order to provide a basis for questions to Vendor A personnel. The resulting schematic is shown in Figure A-4b. The decoding tree/PLA has been broken into sections. For inherent and relative instructions, the net result is the same. All eight lines are required to define an instruction and all of these instructions must be used for a complete test. These instructions are the 66 op codes from 00 to 5F and can be identified as a group by the three higher order bits. The difference occurs for those instructions which are address modifiable (i.e. indexed, extended, immediate and direct). The test now requires the verification of all 66 inherent and relative instructions, all 40 address modifiable instructions and each of the four address modes.

When this mechanization (Figure A-4b) was presented to Vendor A personnel, they stated that the actual mechanization is based on this idea and that the op codes used are sufficient to exercise the instruction decode section.

The Vendor A vector set may be deficient in certain instances with respect to another previously mentioned criteria. That is, test data which is stored in a register must be brought to external pins to determine if an instruction was executed properly; e.g., Vector 431 (Table A-3) is the only time that EORA (EXCLUSIVE OR A with memory) was used. The results of this operation are in the A Register and the ZERO indicator. Both are destroyed (Vectors 467 and 433 respectively) before the results are examined. Further analysis shows that the TSTA in Vector 433 is never verified. There may be other occurrences of op codes whose effect is not brought to the outside pins.

The vectors of Test Set II received from Vendor A are reported to be a more effective test in this area since results of past failure analysis were included. This part of the evaluation was therefore terminated until the effectiveness of the new vector set was determined. Ultimately the new vector set was superior in this area and was completely evaluated.

Vendor A personnel were also asked to explain their use of op codes 00 and 03 which do not appear in their list of instructions. They stated that there are four no-op's which are: 00, 01, 02 and 03. However 00, 02 and 03 may under certain conditions cause unexpected operations. Therefore, 01 is the only recommended NOP for general programming use. Vendor A used 00 and 03 because certain testable areas, unspecified to us, could be tested in fewer vectors. They stated that these same areas could be tested using their listed instructions, however, it would require more test vectors. Since Vendor A did not specify which areas were covered using the no-op's and without a logic diagram it is not possible to define the tests for these areas. This is the intent of Built In Test Electronics (BITE), and it appears that Vendor A is taking advantage of this type of circuitry. Since the existence of this circuitry is not published and a logic diagram is not available, the effects of these op codes cannot be evaluated nor could anyone else take advantage of their existence as BITE. Other than these, Vendor A stated that no other unlisted op codes exist.

Arithmetic and Logic Unit

While evaluating the effectiveness of the vectors on the ALU sector, it was determined that the ALU sector should be further subdivided. Based on these observations and conversations with Vendor A personnel, the diagram in Figure A-5 was developed for evaluation. The significant factor in this diagram is that the partition labeled ALU only performs four functions: ADD, AND, OR and EXOR. The remaining arithmetic and logic functions are implemented with a ones complementor (eight-bit EXOR) on the input lines and a shifter on the output lines. This is significantly simpler than an equivalent 54181 and therefore more straightforward with respect to evaluation and generation of tests. In addition, it was observed that the ALU could operate on both accumulators and return the result to Accumulator A in two cycles. Since the second cycle is required to load Accumulator A with the result, only one cycle is available for transfer of the data from each accumulator to the appropriate part of the ALU. This indicates that two separate data buses are used in this operation and are indicated in Figure A-5 as Buses c and f. Using similar reasoning on memory to accumulator operations, Buses b and d were added.

The test philosophy normally applied to an arithmetic adder/subtractor whose mechanization is not known is to apply all possible input combinations to each bit. That is, for each mode, add and subtract:

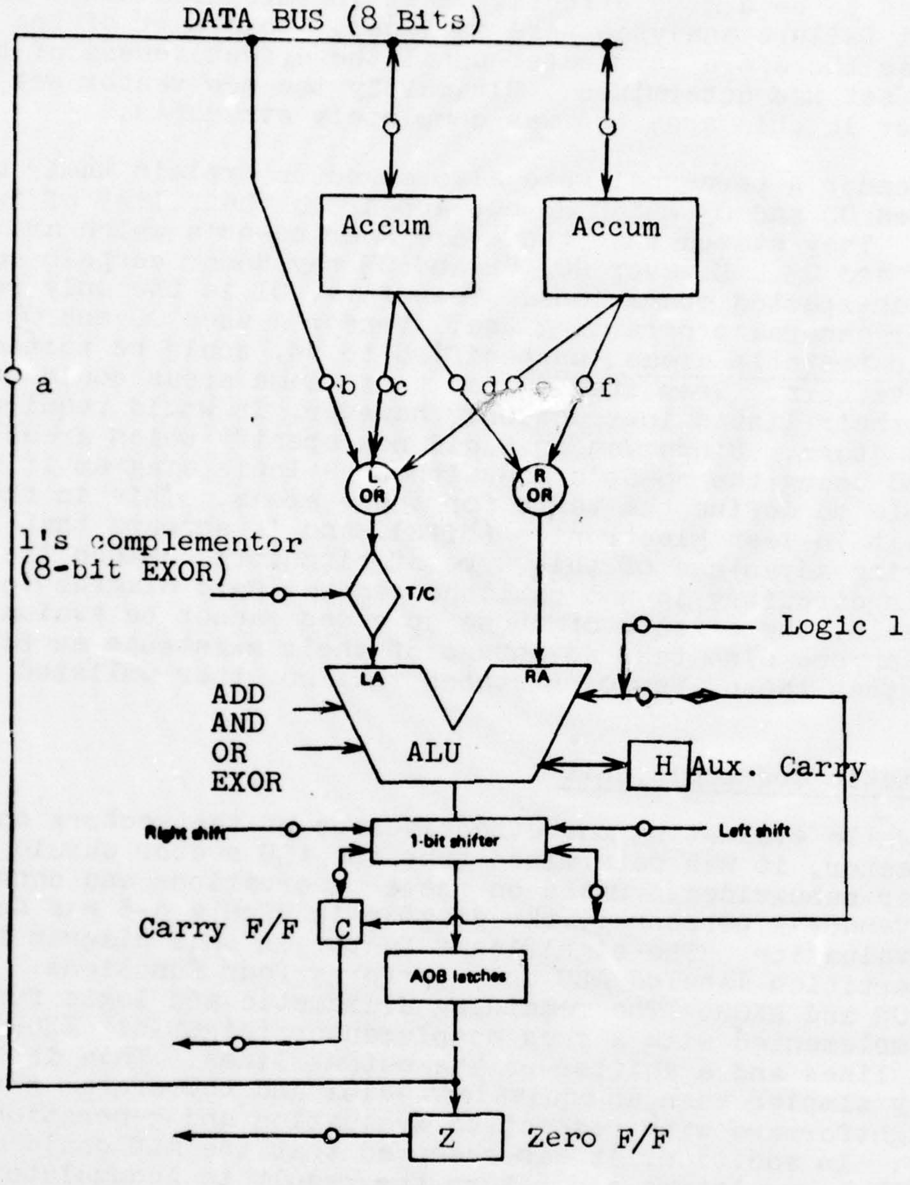


Figure A-5. Sub-sectoring for ALU and Condition Code Register

1) Apply all possible inputs (0 & 0, 0 & 1, 1 & 0 and 1 & 1) to each adder input pair with its carry-in a "zero".

2) Apply all possible inputs to each adder input pair with its carry-in a "one".

In the Mc6800 subtraction is effected by complementing the subtrahend and adding with carry-in equal to "one", that is, converting the subtrahend to a two's complement number and adding. The two's complement of a number is generated by inverting each bit of a number and adding one.

Examples:

Sign							LSB		
0	0	0	0	1	1	0	1	=	+13
1	1	1	1	0	0	1	1	=	-13

For an eight-bit number the maximum signed numbers are:

Sign									
0	1	1	1	1	1	1	1	=	+127
1	0	0	0	0	0	0	0	=	-128

The sign of a number is determined by the state of B7, "zero" = positive and "one" = negative. The remaining bits indicate the magnitude for a positive number, and the two's complement representation of the magnitude for a negative number. The necessary tasks are therefore to:

1) verify the eight possible inputs to each bit of the adder,

2) verify that the complementing circuitry will complement both a "one" and a "zero" for each bit,

3) check bit independence and

4) verify decimal adjust circuitry.

The test philosophy normally applied to the ALU for logic operations is to:

1) Apply the following input conditions to each input pair:

0 & 0, 0 & 1, 1 & 0 and 1 & 1	during EXOR operations
0 & 0, 0 & 1 and 1 & 0	during OR operations
0 & 1, 1 & 0 and 1 & 1	during AND operations

2) Check that for shift left and for shift right operations both "0" and "1" are shifted from each bit into a "0" and a "1" in each adjoining bit. Therefore, four shift left and four shift right combinations are recommended for each bit.

In order to determine that the proper conditions are applied to each input pair of the ALU (for ADD, Subtract, OR, AND, and EXOR), charts (Tables A-1 through A-7) were made which specify any conditions applied without considering sensitivity of the outputs to these conditions. However, in order for these conditions to be an effective test, the results must be sensitized to the outputs. An initial review showed (Refer to Table A-6) that the 1 & 1 conditions for the EXOR were applied only once, and that the results (contained in the A Register) were destroyed before they were made sensitive to an output. Vendor A's test set frequently applies several arithmetic and logic operations to the contents of a register without examining the contents between operations. This test technique makes it impossible or extremely difficult at best to determine sensitive paths without a logic diagram and a fault simulation program. The major difficulty arises when a single fault influences more than one arithmetic and/or logic operation. By concatenating operations without examining the contents of the register in between operations, a failure from one instruction can be masked by succeeding operations which use the same faulted circuitry.

Tables A-1 through A-7 indicate all the conditions that were applied to the ALU and condition code registers. Note that there were many unapplied input conditions. The conditions of Table A-5, A-6 and A-7 were checked as having been sensitized to outputs for verification. The ADD, AND, OR, EXOR operations were not verified. This verification will be performed on Vendor A's new vector set. If these are found to be deficient as was the 1 & 1 EXOR condition in this vector set, they will be augmented at that time.

Input Conditions			Bit Position							
Carry	DI_1	DI_2	7	6	5	4	3	2	1	0
0	0	0	X	X	X	X	X	X	X	X
0	0	1	X	X	X	X	X	X	X	X
0	1	0	X			X	X	X	X	X
0	1	1	X	X	X	X	X	X	X	X
1	0	0	X	X	X	X	X	X	X	
1	0	1	X	X	X	X	X		X	X
1	1	0	X	X	X	X	X	X	X	X
1	1	1	X	X			X	X	X	

Carry Flip-Flop	
0	
1	X

X = condition applied (not necessarily verified)

Blank = condition not applied

Table A-1. Conditions Applied to ALU in ADD Mode

Input Condition		Bit Position							
DI_1	DI_2	7	6	5	4	3	2	1	0
0	0	X	X	X	X	X	X	X	X
0	1	X			X	X	X	X	X
1	0			X					
1	1	X	X	X	X				

Table A-2. Conditions Applied to ALU in OR Mode

Input Condition		Bit Position							
DI_1	DI_2	7	6	5	4	3	2	1	0
0.	0		X	X	X				X
0	1		X	X		X		X	
1	0	X			X	X	X	X	X
1	1	X					X		

Table A-3. Conditions Applied to ALU in AND Mode

Input Condition		Bit Position							
DI_1	DI_2	7	6	5	4	3	2	1	0
0	0								
0	1	X		X		X		X	
1	0		X		X		X		X
1	1	X	X	X	X	X	X	X	X

Table A-4. Conditions Applied to ALU in EXOR Mode

Input Condition	Bit Position							
	7	6	5	4	3	2	1	0
0	X	X	X	X	X	X	X	X
1	X	X	X	X	X	X	X	X

Table A-5. Conditions Applied to the One's Complementor

	Bit Position								Shift-In					Carry Out	
									Force a	Carry-In		ARITH SHIFT			
Transition	7	6	5	4	3	2	1	0	0	0	1	0	1	0	1
0 to 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 to 1		X	X	X	X	X	X	X							
1 to 0	X		X	X	X	X	X	X							
1 to 1	X	X						X							

Table A-6. Conditions Applied to Shift Right

	Bit Position								Carry Out		Force a	Carry-In	
Transition	7	6	5	4	3	2	1	0	0	1	0	0	1
0 to 0	X	X	X	X	X	X	X	X	X	X	X	X	X
0 to 1	X			X	X	X	X	X					
1 to 0	X	X		X	X	X	X	X					
1 to 1	X	X	X					X					

Table A-7. Conditions Applied to Left Shift

Register Array

The Register Array in the Mc6800 consists of the following registers:

1) Two 8-bit accumulators - These are the most powerful registers and can be used in any arithmetic or logic instruction and for temporary data storage.
2) One 16-bit Index Register used for indexed address modification and temporary data storage.
3) One 16-bit Stack Pointer - This is a special purpose register used to identify the area of RAM where the other registers are stored during an interrupt or subroutine.
4) One 16-bit Program Counter - This is a specialized register which points to the next instruction to be executed at each point in a program.
5) A 6-bit Condition Code Register which stores the Half-Carry, Interrupt Mask, Negative, Zero, Overflow and Full-Carry conditions from previous operations and is used in Conditional Branch operations.
6) Two 8-bit temporary registers associated with the data and address buses.
7) And also an Incrementor/Decrementor which operates on the three 16-bit registers and the address buffer latches.

A register array is a random access memory and, as such, lends itself to tests using many of the standard tests developed for RAMs. For example, it is possible to apply modified Walking 1/0 and Galloping 1/0 patterns to most of the register array. The Vendor A test set does not include such patterns. The designers of the Mc6800, who also developed the test vectors, were contacted and questioned as to why these test were not included. They explained that the register arry was purposely built out of static latches over a large area of the chip in order that the pattern and temperature sensitivities associated with RAMs would be eliminated. (Note: Vendor A specifies a minimum clock frequency. This is because the buses were designed to operate dynamically, i.e. they depend on capacitance for data transfer.) Based on this assumption, the evaluation approach was to check for:

1) register independence,
2) bit independence and
3) flip-flop integrity (i.e. insure transitions of 0 to 0, 0 to 1, 1 to 1 and 1 to 0 for each bit in each register).

The registers are uniquely specified several times. This occurs whenever an interrupt occurs and the data in each 8-bit section of the register array, (i.e. each single-length register and each half of the double-length registers) is unique. An example of this occurs at Vector 192 in the long test (Table A-3). The registers were previously loaded with the following:

Register	Data
PC (Low)	C9
PC (High)	OD
INDX (Low)	OO
INDX (High)	96
A	81
B	11
Flag	FO

When the op code SWI (Software Interrupt) is executed, the contents of the registers are sequentially output to the data bus. If the register select circuitry is faulty, the correct data will not be output in the correct order.

The flip-flops in the A, B and Program Counter Registers are shown independent. Adjacent Bits 6 and 7 in the Stack Pointer are not proven independent and several pairs of flip-flops were not shown independent in the Index Register. Flip-flop integrity was examined for each of the registers. The results are detailed in Tables A-8 through A-13.

In summary, tests should be added to complete the verification of flip-flop independence and integrity.

States & Transitions	Bit Positions							
	7	6	5	4	3	2	1	0
Data = 0	X	X	X	X	X	X	X	X
Data = 1	X	X	X	X	X	X	X	X
0 to 0	X	X	X	X	X	X	X	X
0 to 1	X	X	X	X	X	X	X	X
1 to 0	X	X	X	X	X	X	X	X
1 to 1	X	X	X	X	X	X	X	X

X = Test Accomplished

Blank = Test Not Accomplished

Table A-8. Flip-Flop Integrity for the A Register

States &	Bit Positions							
Transitions	7	6	5	4	3	2	1	0
Data = 0	X	X	X	X	X	X	X	X
Data = 1	X	X	X	X	X	X	X	X
0 to 0	X	X	X	X	X		X	X
0 to 1	X	X	X	X	X	X	X	X
1 to 0	X	X	X	X	X	X	X	X
1 to 1		X	X	X	X	X	X	X

Table A-9. Flip-Flop Integrity for the B Register

States &	Bit Positions					
Transitions	H	I	N	Z	V	C
Data = 0	X	X	X	X	X	X
Data = 1	X	X	X	X	X	X
0 to 0		X	X	X	X	
0 to 1	X	X	X	X	X	X
1 to 0	X	X	X	X	X	X
1 to 1		X	X	X	X	X

Table A-10. Flip-Flop Integrity for the Condition Code Register

States &	Bit Position															
Transitions	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data = 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Data = 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 to 0	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X
0 to 1	X		X	X		X	X	X	X	X	X	X	X	X	X	X
1 to 0	X			X		X	X	X	X	X	X	X	X	X	X	X
1 to 1	X		X	X		X	X		X	X	X	X	X	X	X	

Table A-11. Flip-Flop Integrity for the Index Register

States &	Bit Position															
Transitions	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data = 0		X	X	X	X	X	X		X	X	X	X	X	X	X	X
Data = 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 to 0			X		X		X		X	X	X	X	X	X	X	
0 to 1		X			X	X	X					X	X	X	X	X
1 to 0		X	X	X		X			X	X	X	X	X	X	X	X
1 to 1	X			X				X	X	X	X	X	X	X	X	

Table A-12. Flip-Flop Integrity for the Stack Pointer Register

States & Transitions	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data = 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Data = 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 to 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0 to 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 to 0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 to 1	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table A-13. Flip-Flop Integrity for the Program Counter

SECTION A.2

EVALUATION OF A TEST PROCEDURE FOR LEAKAGE CURRENT MEASUREMENTS FOR THE Mc 6800 MICROPROCESSOR

Control Input Leakage Tests

Vendor A's control input leakage tests are oriented toward surface leakage since the tests are run in a powered down mode. The test is performed with all pins grounded except the pin under test, which is returned to 5.25 volts. A test condition indicative of operating-worse case leakage was attempted as a replacement for the Vendor A test. The results are as follows:

1) In a powered up mode the device is unpredictable if it is statically operated. The immediate results are to cause the chip to go into modes of operation that draw large power supply currents. In the cases that were experienced, Vendor A stated that the output prebuffers were in a power-hogging state. This can cause stress conditions and reliability problems with the device. The device is truly dynamic in that internal chip reference supplies are dependent upon the $\emptyset_1$, $\emptyset_2$ clocks. This means that low-level leakage tests would have to be performed dynamically in a powered-up mode.
2) A worse case test condition can only be arrived at if a logic diagram and circuit schematic are available. In conversations with Vendor A, they indicated that a topological map would be necessary also to take advantage of the proximity of certain gates relative to the input under test.
3) Vendor A is satisfied with the present test.

The conclusion is that although the leakage measurements are not truly indicative of operating leakage levels it is the only realistic alternative. Since measurement settling times are necessary for low level current measurements, a dynamic test would prove highly impractical, if not impossible, since the chip clocks cannot be run any slower than 100KC. Also chip schematics and diagrams are not obtainable.

Preconditioning for Tri-state Output Leakage Tests

*The preconditioning pattern is an orderly procedure that is used to establish worse case conditions on the address and data outputs when performing the tri-state leakage tests. There are two preconditioning patterns, one for the low level leakage test and one for the high level leakage test.

Some of the reasoning for the sequence of the instructions, in particular the use of the unidentified no-ops. is obscure because of the availability of information from Vendor A. Verification of the worse case test goals, however, is still possible if the test pattern is entered into and actually performed on the uP.

*NOTE: The preconditioning pattern is detailed in MIL-M-38510/400.

The preconditioning pattern sets up the worst case circuit situation for measuring the high impedance state leakage current. Since there are basically two leakage tests made to a tri-state output (leakage current - forcing voltage high, 2.4V; leakage current - forcing voltage low, 0.4 volt), it also follows that there are two worst case circuit situations. These are achieved by exercising the device with two different patterns. The goal is to cause a logic "0" to be applied to the input of the tri-state buffer element within the microprocessor when the forcing voltage for the leakage measurement is high (2.4V). This causes the worst case potential to be applied across the output buffer for this test. Similarly a logic "1" should be applied to the buffer element input when the forcing voltage is low (.4V). These conditions are shown in Figure A-6.

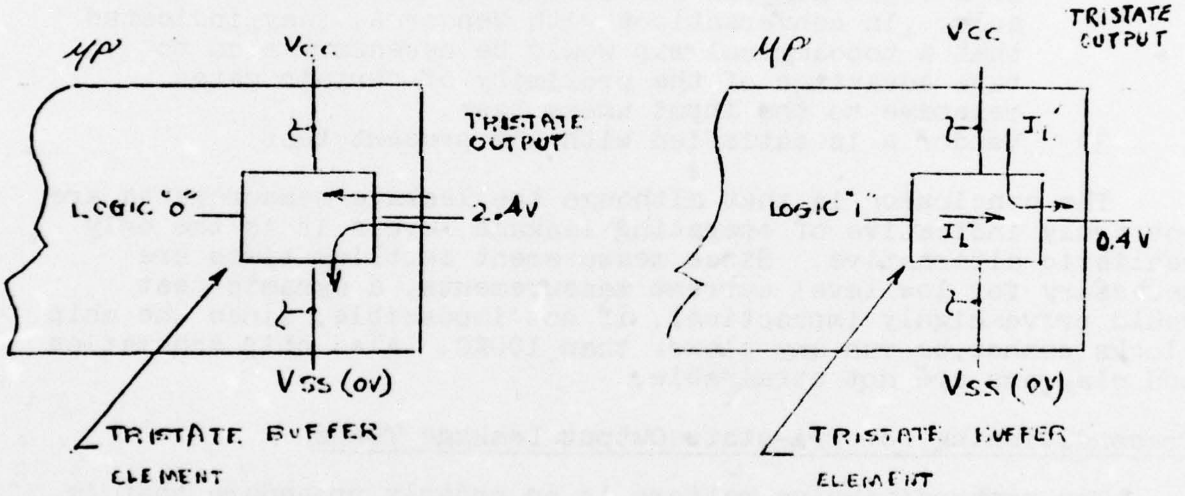


Figure A-6. Tri-state Output Leakage

Vendor A indicated that this worst case condition was applied to the address and data lines.

The preconditioning pattern causes the device to enter the worse case mode of operation for the tri-state leakage tests; however, this mode cannot be retained indefinitely. Although the device is being clocked, while in the looping pattern, it can only retain this looping mode of operation for approximately five milliseconds. Vendor A stated that there is an internal V_{gg} reference supply that is generated internal to the device. This reference runs off the clocks and degrades as a function of time in this mode.

One main change made to the Vendor A test was to force all other data and address outputs high when measuring low level leakage on an output and similarly force all other data and address outputs low when measuring high level leakage current. This will add several additional leakage paths to each test. The test procedure is outlined below.

Test Procedure

1) Perform preconditioning test pattern.
2) Loop on the last two vectors (81 and 82).
3) While in the looping pattern, apply the test voltages to the device pins (tri-state current low).
4) Repeat the preconditioning pattern and loop on Vectors 81 and 82.
5) While looping, perform the leakage test on the next output pin. Continue to repeat this procedure until all tri-state low level leakage currents on all pins are complete.
6) Perform the preconditioning except that Vectors 36 and 38 should be changed as indicated.
7) Loop on Vectors 81 and 82. While in the looping pattern, apply the test voltages to the device pins, (tri-state current high).
8) Repeat Steps 6 and 7 for each successive test step until all output pins have been tested.

During the preconditioning, monitoring of the address and status outputs is not done or required; when the data buses are in the output mode, it is also not monitored. Their status is, therefore, not included in the preconditioning table.

The waveforms applied to the device during preconditioning and while in the loop are shown in Figure A-7.

Following is an explanation of what the preconditioning test pattern is actually doing. Note that each test vector represents one half of a clock cycle in that both Ø1 and Ø2 clocks are required to complete a chip cycle. Alternate vectors are assigned to Ø1 and Ø2 clock states.

Test Vectors

1 - 20 The device is halted ($\overline{\text{HALT}} = 0$), and the data bus is disabled ($\overline{\text{DBE}} = 0$). Ten full clock cycles are applied to the device prior to any software or hardware commands.

21 - 26 The interrupt request input goes high and the halt input low. This starts an interrupt sequence, but it is never allowed to finish. A restart occurs after three cycles where a hardware interrupt requires better than nine cycles. It is not known what the microprocessor is actually doing internally at this point.

27 - 34 The reset input goes high and a reset sequence is initiated on the first clock cycle. The next two cycles force the high and low address outputs high (FFFE, FFFF). Normally a stored initializing program that sets up system starting conditions would start in this memory location. The next cycle reads the op code 8E (hexadecimal) which is the designation for load stack pointer. The uP thinks this is being retrieved from memory. This is a three-cycle command.

35 - 38 The next two cycles load the higher and lower order addresses for the Stack Pointer. The address is comprised of all ones which precondition the inputs to the address output buffers for the output-low leakage tests. For the output-high leakage tests the preconditioning pattern is repeated except the Stack Pointer address is changed to zeros to again provide a worse case condition.

39 - 58 A return from interrupt command is now read (3B). This instruction is one byte long and requires ten cycles. The first two cycles include reading the instruction and performing an internal cycle. Succeeding cycles perform the following:

increment the Stack Pointer
restore the Condition Code Register
restore Accumulator B
restore Accumulator A
restore Index Register high
restore Index Register low
restore Program Counter high
restore Program Counter low

When Accumulator B was restored, (Vectors 47, 48) logic ones were read in from the data inputs. This causes the worse case condition on the data outputs

Test Vectors

39 - 58 (Continued) for the low level leakage current test. In a system application this restored data would have been stored in memory and later accessed for the return from interrupt.

59 - 76 The next instruction is wait for interrupt (3E). This is one byte long but requires nine clock cycles. This essentially takes all of the data that was restored in the return from interrupt and makes it available on the data bus but does not destroy it.

77 - 80 The wait for interrupt is followed by a no-op (no operation command) that is executed twice. The performance of the no-op has not been defined by Vendor A and is not recorded as a usable instruction in their literature. It is usable, however, in the respect that Vendor A can predict the uP response when it succeeds certain instructions, in this case the wait for interrupt. Vendor A was able to determine that the wait for interrupt followed by the two no-ops cause the desired response from the uP. This cannot be verified by checking the circuitry because of the unavailability of information on the no-ops, but can be verified by actually performing the test.

81 - 82 The data bus enable and tri-state control inputs are at a logic "0" and "1" respectively which cause the data and address lines to enter the high impedance state. Since the Ø2 clock is used for data transfers, only the Ø1 clock is required. The data that was previously entered for the worse case condition for the data output leakage tests is stored and does not depend upon the Ø2 clock.

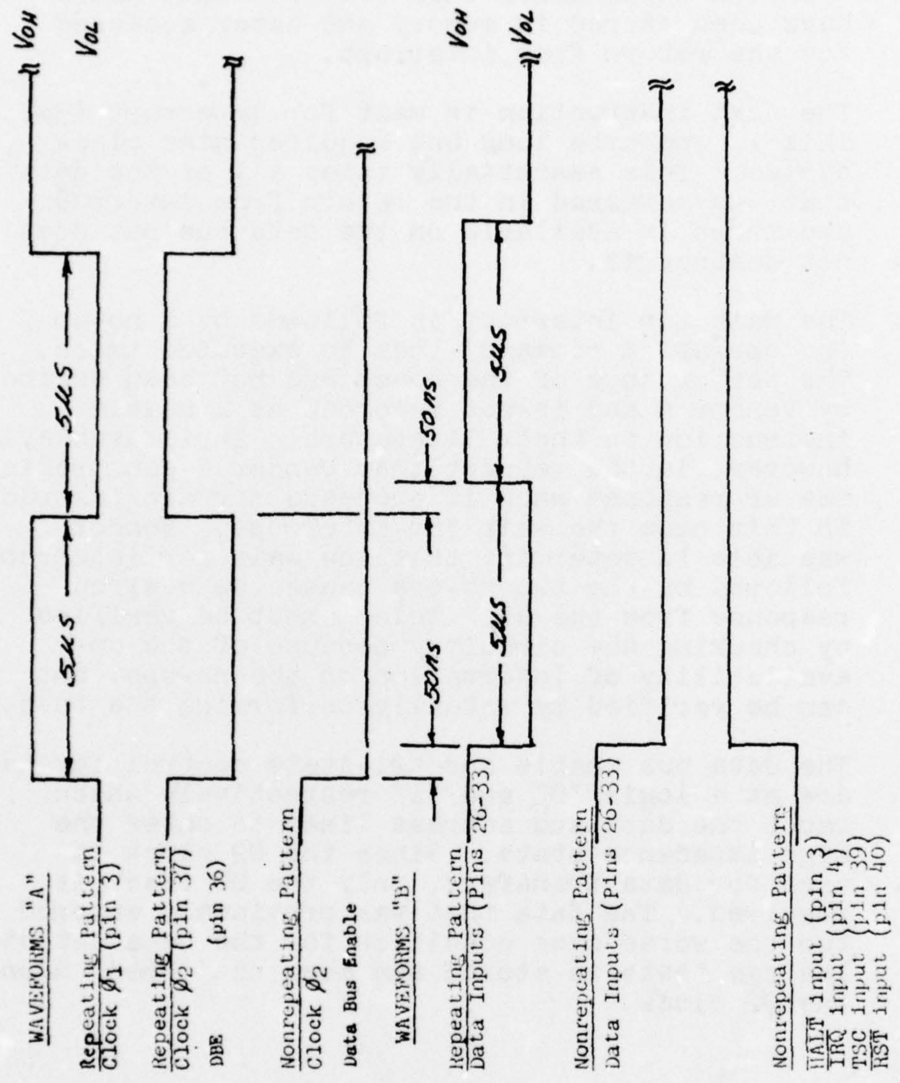

VOH
VOL
5µs
5µs
50ns
50ns
5µs
5µs
WAVEFORMS "A"
Repeating Pattern Clock Ø1 (pin 3)
Repeating Pattern Clock Ø2 (pin 37)
DBE (pin 36)
Nonrepeating Pattern Clock Ø2
Data Bus Enable
WAVEFORMS "B"
Repeating Pattern Data Inputs (Pins 26-33)
Nonrepeating Pattern Data Inputs (Pins 26-33)
Nonrepeating Pattern
HALT input (pin 3)
IRQ input (pin 4)
TSC input (pin 39)
RST input (pin 40)

Figure A-7. Tri-state Preconditioning Pattern Waveforms

NOTES:

1. A repeating pattern is one which alternately changes state from vector to vector.
2. Clock phases 1 and 2 are nonoverlapping; $\emptyset_1$ and $\emptyset_2$ cannot be at a logic "1" simultaneously.
3. Tr = tf = 5 ns min., 50 ns max. measured between 10-90%.
4. Timing delays measured at 50% of V_{OH}, V_{OL}.
5. V_{OH} = 2.4V, V_{OL} = 0.4V.

APPENDIX B

B.1 EVALUATION OF A FUNCTIONAL TEST SET FOR THE VENDOR B 8080 MICROPROCESSOR

INTRODUCTION

The functional test program that was evaluated was written by an automatic tester manufacturer. The results of the evaluation would determine if this program or one similar to it could be used as a basis for a MIL-M-38510 slash sheet for the Vendor B 8080 Microprocessor. It would also provide for comparison purposes another approach to testing microprocessors.

Because of the complexity of the I8080 and other VLSI (Very Large Scale Integration) devices, the development and use of all input vectors for each state of the microprocessor would be impossible for all practical purposes. Therefore, for either test generation or test evaluation, a model representing the internal functional blocks of the uP (microprocessor) must be used which identifies the internal machine states and interconnecting data paths. This should include both a gate-level logic diagram, timing diagrams and a block diagram which shows the major functional areas of the uP and the interconnecting data and control paths. Since neither a logic diagram nor a sufficiently detailed block diagram were available, the approach taken to evaluate the effectiveness of the functional test was to:

1) Section the uP into functional blocks (refer to Figure A-8) and develop a block diagram based on available information.
2) Analyze the test vectors to determine what rationale was used and its validity.
3) Based on experience, evaluate the effectiveness of the vectors in c omparison with the type of tests required for each functional block. In order to evaluate the test vectors more effectively and efficiently, a computer program was written to convert the binary representation of the vectors to an assembler language representation. This program will be referred to in this report as the "disassembler".
4) Verify that the interconnecting data and control paths were checked.
5) Ensure that the instruction set is verified.

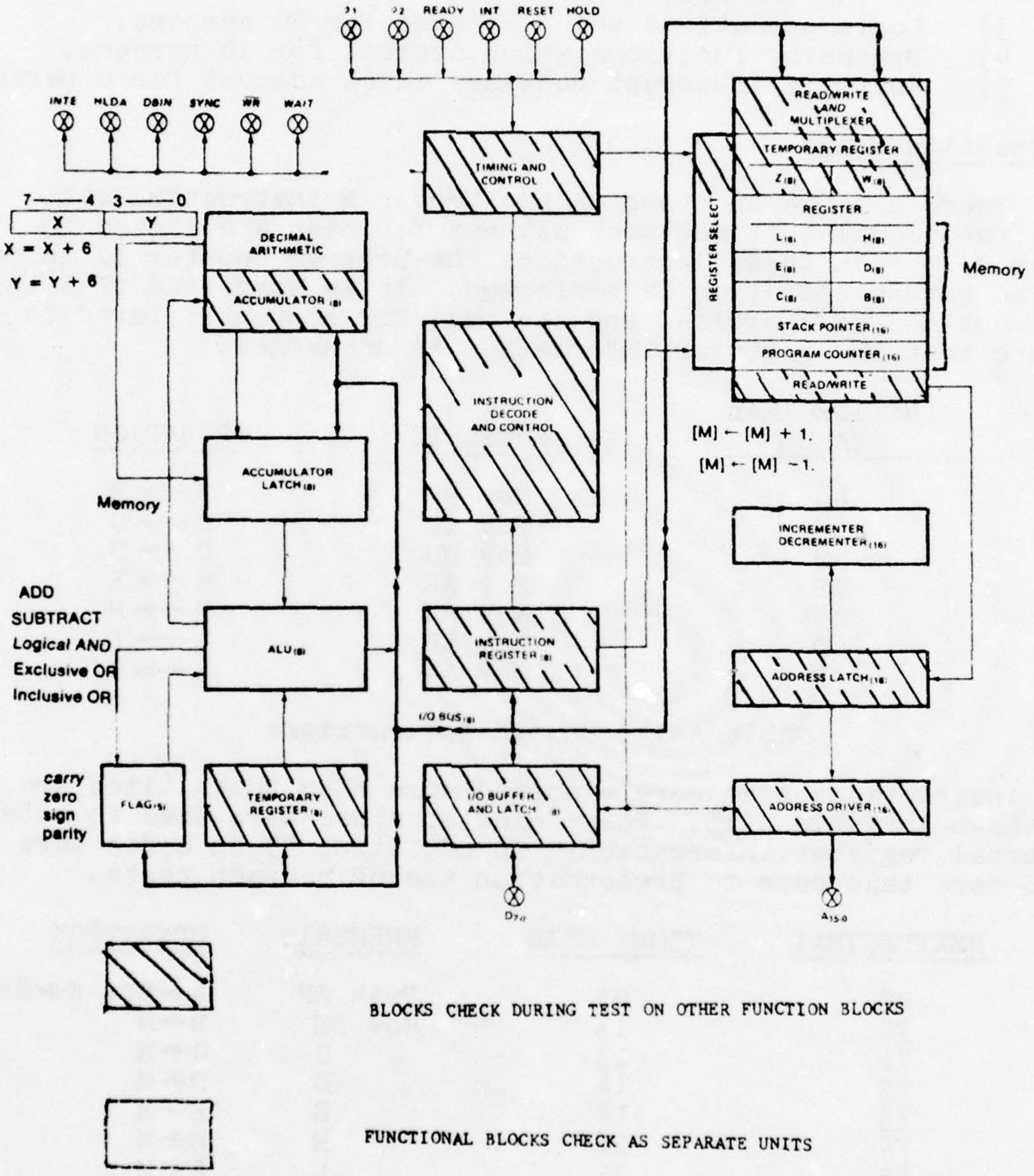


Figure A-8. I8080-A Functional Block Diagram

DISCUSSION

The functional test program that was evaluated can be divided into five parts. They are:

1) Movement of data between registers which comprises 43 percent of the test vectors.
2) Arithmetic operations which account for 15 percent of the vectors.
3) Logic operations which account for 20 percent.
4) Branching functions which account for 16 percent.
5) Software interrupt routines which account for 6 percent.

Instruction Usage

There are 224 op codes in the Vendor B instruction set. This functional test set uses all but 7, which are listed in Table A-14. In these instructions the program counter is incremented but no operation is performed. It is suggested that these instructions be exercised and the register contents listed to ensure that the register information is unchanged.

HEXIDECIMAL VALUE	MNEMONIC	OPERATION
40	Mov BB	B → B
49	Mov CC	C → C
52	Mov DD	D → D
5B	Mov EE	E → E
64	Mov HH	H → H
6D	Mov LL	L → L
7F	Mov AA	A → A

Table A-14. Unused Instructions

All instructions that were executed more than three times are as shown in Table A-15. These nine op codes were used to obtain internal register information. In addition, 25 op codes were used more than once to precondition the uP between tests.

HEXIDECIMAL	TIMES USED	MNEMONIC	OPERATION
F5	90	Push AF	A → SP; F → SP
70	13	Mov BM	B → M
71	13	C	C → M
72	14	D	D → M
73	14	E	E → M
74	21	H	H → M
75	21	L	L → M
77	12	A	A → M
C9	10	Ret	Return from subroutine

Table A-15. Most Commonly Used Instructions

It should be noted that no instruction was used twice consecutively.

Register Increment, Decrement and Load Operation

This portion of the test walks a "one" through a field of "zero's" in each register. However, the movement across the register is not complete. The following bit locations were never written with a one:

1) Register B Bit_4, Bit_6, Bit_7
2) Register C Bit_7
3) Register D Bit_2

Walking a "zero" in a field of "one's" is never attempted.

All special operations on the registers are performed, increment, decrement and revise registers.

Table A-16 presents the status of each register after each functional test sequence, the test vectors included in each sequence and the operations required to obtain the resultant status.

Data written into one register may disturb the contents of an adjacent cell. Vendor B considers that the area of the register is so large the adjacent cell disturbance tests are not required. This test set, however, tests this area by performing a pseudo-walking one. All registers are loaded with a single bit at a logic "1" in a field of zeroes; and after each test sequence, the registers are checked for disturbance. Note (refer to Table A-17) that the movement of register data also forms a partial diagonal "1" pattern. Many manufacturers consider this a more effective test pattern than a walking one pattern since it checks the rows and columns for uniqueness. This pattern could have been completed if more instructions were repeated.

Table A-18 delineates (with a one) those register bits that were tested with a single "one" in a field of "zero's". An X indicates bits tested with a "one" but not in a field of all "zero's". There was no testing for a single "zero" in a field of "one's".

The Bit positions not shown with 1 or X were never loaded with a one during this test.

Experience has shown that, at a minimum, every register cell should be tested by writing a "one" over a "zero" and a "zero" over a "one" to ensure proper register operation. In addition, certain mechanizations require a "one" to be written over a "one" and a "zero" over a "zero" to obtain a 100 percent TCL.

Since Vendor B register mechanization is not known, at a minimum, additional vectors should be added to complete verification of bit independence and to assure that for every bit location a "one" is written over a "zero" and a "zero" over a "one". With the addition of these vectors and assuming internal disturbance is not a problem, a sufficient test is performed. If it is determined that sensitivities occur, additional test vectors would be added.

Table A-16. Register Contents

X = No Change

DATA BUFFER MEMORY SEQUENCE	A	B	C	D	E	H	L	SP_L	SP_H	OPERATION
33 to 68	64	1	2	4	8	16	32	0	254	Load B-C, D-E, SP, A, Store A Load A-Flags, Store A-Flags, B-C, D-E, H-L
69 to 92	128	2	4	8	16	32	64	X	X	Load B-C, D-E, H-L Store A, B, C, D, E, H, L
93 to 98	X	X	X	X	64	8	16	X	X	Reverse D-E and H-L Store D, E, H, L
99 to 109	X	X	X	X	X	0	254	8	16	Reverse SP and H-L Store H, L
110 to 119	X	X	X	X	X	1	252	0	254	Load H-L into SP & PC, Add SP H-L Store and push (check address)
120 to 124	X	X	X	X	X	4	0	X	X	Add B-C to $\overline{HL}$ Store H, L
125 to 129	X	X	X	X	X	36	64	X	X	Add D-E to $\overline{HL}$ Store H, L
130 to 142	0	X	X	X	X	72	128	X	X	Add H-L to $\overline{HL}$, Load A Store H, L, B-C, D-E, Load A, Store A
143 to 165	254	X	5	X	65	X	129	X	255	Add one to B-C, D-E, H-L, SP Store H-L, B, C, D, E, H, L
166 to 184	X	X	4	X	64	X	128	X	254	Sub one for B-C, D-E, H-L, SP Store H-L, B, C, D, E, H, L
185 to 196	(0)^1 157^2	X	X	X	X	X	X	X	X	^{1}Complement A ^{2}Load A

Table A-16. Register Contents (Continued)

X = No Change

DATA BUFFER MEMORY SEQUENCE	A	B	C	D	E	H	L	SP_L	SP_H	OPERATION
197 to 200	3	X	X	X	X	X	X	X	X	Decimal adj A Store A
201 to 204	105	X	X	X	X	255	0	X	X	Decimal adj A Store A Store H-L, Load H-L, Store H-L
205 to 256	1	2	4	8	16	32	64	X	X	Load A, B, C, D, E, H, L Store A, B, C, D, E, H, L
257 to 284	2	4	8	16	32	64	128	X	X	Load A, B, C, D, E, H, L Store A, B, C, D, E, H, L
285 to 308	3	5	9	17	33	65	129	X	X	Increment A, B, C, D, E, H, L Store A, B, C, D, E, H, L
309 to 329	2	4	8	16	32	64	129	X	X	Decrement A, B, C, D, E, H, L Store A, B, C, D, E, H, L
330 to 350	4	8	16	32	64	108	4	X	X	A←B, B←C, C←D, D←E, E←H, H←L, L←A Store A, B, C, D, E, H, L
351 to 371	16	32	64	128	4	16	32	X	X	A←C, B←D, C←E, D←H, E←L, Store A, B, C, D, E; H←A, L←B, Store H, L
372 to 383	128	4	16	32	X	X	X	X	X	A←D, B←E, C←H, D←L, Store A, B, C, D, E←A; H←B, L←C, Store E, H, L
384 to 392	X	X	X	X	128	4	16	X	X	E←A, H←B, L←C, Store E, H, L
393 to 419	64	8	2	64	4	64^1 16^2	2^1 32^2	X	X	Load H, L A←E, E←B, B←H, H←C, C←L, L←D, D←A Store A, B, C, D, E, H, L
420 to 440	16	32	16	8	2	64	4	X	X	A←H, H←D, D←B, B←L, L←E, E← C, C←A, Store A, B, C, D, E, H, L
441 to 461	4	4	32	16	8	2	64	X	X	A←L, L←H, H←E, E←D, D←C, C←B, B←A, Store A, B, C, D, E, H, L

Test Vector	X = No Change							
Register	329	350	371	383	392	419	440	461
A	A	B	C	X	X	E	H	L
B	B	C	D	E	X	H	L	A
C	C	D	E	H	X	L	A	B
D	D	E	H	L	X	A	B	C
E	E	H	L	X	A	B	C	D
H	H	L	X	A	B	C	D	E
L	L	A	X	B	C	D	E	H

Table A-17. Register Content Movement

	REGISTER						
POSITION	A	B	C	D	E	H	L
0	1	1	X	X	X	1	X
1	1	1	1		1	1	1
2	1	1	1	1	1	1	1
3	X	1	1	1	1	1	X
4	1		1	1	1	1	1
5	X	1	1	1	1	1	1
6	1		1	1	1	1	1
7	1			1	1	1	1

1 = Register bits that were tested with a single "1" in a field of "zero's".

X = Register bits that were tested with a single "1" in a field of "one's" and "zero's".

Table A-18. Register Bit Pattern

Arithmetic Operations

The test philosophy normally applied to an arithmetic adder/subtractor whose mechanization is not known is to apply all possible input combinations. That is,

1) Apply all possible inputs (0 & 0, 0 & 1, 1 & 0, and 1 & 1) to each adder input pair with its carry-in a "zero".
2) Apply all possible inputs to each adder input pair with its carry-in "one".

The sign of a number is determined by the state of B7, "zero" = positive and "one" = negative. The remaining bits indicate the magnitude for a positive number, and the two's complement representation of the magnitude for a negative number. The two's complement of a number is generated by inverting each bit of a number and adding one.

Examples:

Sign							LSB		
0	0	0	0	1	1	0	1	=	+13
1	1	1	1	0	0	1	1	=	-13

For an eight bit number the maximum signed numbers are:

Sign							LSB		
0	1	1	1	1	1	1	1	=	+127
1	0	0	0	0	0	0	0	=	-128

Addition or subtraction which produces a number that cannot be represented by a given word length results in overflow. For addition, overflow occurs when the carry-out differs from the carry-in of the sign position.

During all arithmetic and logic routines, the registers are loaded with a single "one" bit in a field of "zero's" (as shown in Table A-19). The registers are then in turn combined with the Accumulator A through the ALU. This sequence results in only one bit changing in the accumulator per instruction.

The operations in the arithmetic logic unit (ALU) are ADD, AND, OR, EXOR, and INVERT. This eight-bit ALU is made up of two "Four-bit Ripple Carry Adders". Each four-bit adder adds four bits of "A" with four bits of "B" and a carry input, resulting in four sum bits and a carry output. There are no controls to the internal ripple carry structure within each four-bit adder. The carry-out of each four-bit adder feeds a flip flop. One four-bit adder feeds the "Carry Y" flip-flop, the other feeds the "auxiliary flip-flop only used during the BCD adjustment.

Register	Register Bits							
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
B	0	0	0	0	0	1	0	0
C	0	0	1	0	0	0	0	0
D	0	0	0	1	0	0	0	0
E	0	0	0	0	1	0	0	0
H	0	0	0	0	0	0	1	0
L	0	1	0	0	0	0	0	0

Table A-19. Conditions of Registers for ALU Routines

The "ADD Without Carry-In" test sequence is shown in Table A-20. Refer to Table A-19 for the status of Registers B through L.

Accumulator Plus Register	Register Bits									
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	Magnitude	Sign Magnitude
A	1	0	0	0	0	0	0	0	= 128	-128
$A + B = A_1$	1	0	0	0	0	1	0	0	= 132	-120
$A_1 + C = A_2$	1	0	1	0	0	1	0	0	= 164	- 88
$A_2 + D = A_3$	1	0	1	1	0	1	0	0	= 180	- 72
$A_3 + E = A_4$	1	0	1	1	1	1	0	0	= 188	- 64
$A_4 + H = A_5$	1	0	1	1	1	1	1	0	= 190	- 62
$A_5 + L = A_6$	1	1	1	1	1	1	1	0	= 254	- 2
$A_6 + I = A_7$	1	1	1	1	1	1	1	1	= 255	- 1
$A_7 + A_7$	1	1	1	1	1	1	1	0	= 254	- 2

Table A-20. ADD Without Carry-In

The following conditions were tested with carry-in to each respective adder = "zero".

Adder Input Pair	Input Condition			
	0,0	0,1	1,0	1,1
0	X	X		X
1	X	X	X	
2		X	X	
3	X	X	X	
4	X	X	X	
5	X	X	X	
6	X	X	X	
7			X	

The following conditions were tested with carry-in to each respective adder = "one".

Adder Input Pair	Input Conditions			
	0,0	0,1	1,0	1,1
0				
1				X
2				X
3				X
4				X
5				X
6				X
7				X

The "ADD with Carry-In" is shown in Table A-21. Refer to Table A-19 for the initial states of Registers B through L.

Accumulator Plus Register Plus Carry-In	Register Bits								Magnitude	Sign Magnitude
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0		
A	1	1	1	1	1	1	1	0	= 254	-2
$A + B + 1 = A_1$	0	0	0	0	0	0	1	1	= 3	+3
$A_1 + C + 1 = A_2$	0	0	1	0	0	1	0	0	= 36	+36
$A_2 + D + 0 = A_3$	0	0	1	1	0	1	0	0	= 52	+52
$A_3 + E + 0 = A_4$	0	0	1	1	1	1	0	0	= 60	+60
$A_4 + H + 0 = A_5$	0	0	1	1	1	1	1	0	= 62	+62
$A_5 + (-62) + 0 = A_6$	0	0	0	0	0	0	0	0	= 0	+0
$A_6 + L + 1 = A_7$	0	1	0	0	0	0	0	1	= 65	+65
$A_7 + A_7 + 0$	1	0	0	0	0	0	1	0	Overflow = 135	-127

Table A-21. ADD With Carry-In

The following conditions were tested with carry-in to each respective adder = "zero".

Adder Input Pair	Input Conditions			
	0,0	0,1	1,0	1,1
0	X			X
1	X	X	X	X
2	X		X	X
3	X	X	X	
4	X	X	X	
5	X	X	X	
6	X	X		X
7	X			

The following conditions were tested with carry-in to each respective adder = "one".

Adder Input Pair	Input Conditions			
	0,0	0,1	1,0	1,1
0	X		X	
1	X		X	
2	X		X	
3			X	
4			X	
5			X	
6		X	X	
7	X	X	X	

Two's complement subtraction without carry-in is shown in Table A-22. Two's complement subtraction is performed by inverting the subtrahend and adding one and adding that result to the minuend. Reference Table A-19 for the states of Registers B through L.

Accumulator Minus Register	Register Bits								Magnitude	Sign Magnitude
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0		
A	1	0	0	0	0	0	1	0	= 130	-127
A-B	0	1	1	1	1	1	1	0	Overflow = 126	+126
A-C	C	1	0	1	1	1	1	0	= 94	+94
A-D	0	1	0	0	1	1	1	0	= 78	+78
A-E	0	1	0	0	0	1	1	0	= 70	+70
A-H	0	1	0	0	0	1	0	0	= 68	+68
A-L	0	0	0	0	0	1	0	0	= 4	+4
A-A	0	0	0	0	0	0	0	0	= 0	+0
A-(1)	0	0	0	0	0	0	0	1	= 1	+1

Table A-22. Two's Complement Subtraction Without Carry-In

The following conditions were tested with carry-in to each respective adder = "zero".

Adder Input	Input Conditions			
Pair	0,0	0,1	1,0	1,1
0	X	X		
1	X		X	X
2	X	X	X	X
3	X	X	X	X
4	X	X	X	X
5	X	X		X
6	X	X		X
7	X			X

The following conditions were tested with carry-in to each respective adder = "one".

Adder Input	Input Conditions			
Pair	0,0	0,1	1,0	1,1
0				
1				
2				X
3		X		
4		X		
5		X		
6		X		X
7		X		

Two's complement subtraction with carry-in is shown in Table A-23. This subtraction is performed by inverting the subtrahend and adding carry-in to it. This result is then added to the minuend. Reference Table A-19 for the state of Registers B through L.

Accumulator Minus Register Minus Carry-In	Register Bits								Magnitude	Sign Magnitude
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0		
A	0	0	0	0	0	0	0	1	= 1	+1
A-B-1	1	1	1	1	1	1	0	0	= 252	-2
A-C-1	1	1	0	1	1	0	1	1	= 219	-34
A-D-1	1	1	0	0	1	0	1	1	= 203	-50
A-E-1	1	1	0	0	0	0	1	1	= 195	-58
A-H-1	1	1	0	0	0	0	0	1	= 193	-60
A-L-1	1	0	0	0	0	0	0	1	= 129	-127
A-125-1	1	1	1	1	1	1	1	1	= 255	-1
A-A-1	1	1	1	1	1	1	1	1	= 255	-1

Table A-23. Two's Complement Subtraction With Carry-In

The following conditions were tested with carry-in to each respective adder = "zero".

Adder Input Pair	Input Conditions			
	0,0	0,1	1,0	1,1
0		X	X	X
1		X	X	
2	X		X	X
3		X	X	
4		X	X	
5		X	X	
6		X	X	
7		X	X	

The following conditions were tested with carry-in to each respective adder = "one".

Adder Input Pair	Input Conditions			
	0,0	0,1	1,0	1,1
0				X
1	X	X	X	X
2		X		
3		X	X	X
4		X	X	X
5		X	X	
6			X	X
7				X

There are four separate vectors used to check add-immediate and subtract - immediate with Byte 2. They are listed in Table A-24.

Immediate indicates that the accumulator is to be combined with the next word to appear on the data lines, Byte 2.

Accumulator Plus Byte 2 (b2)	Register Bits								Magnitude	Signed Magnitude
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0		
A	1	1	1	1	1	1	1	1	= 255	-1
A + $b^{\underline{2}}(1)$	0	0	0	0	0	0	0	0	= 0	0
A + $b^{\underline{2}}(-1)+1$	0	0	0	0	0	0	0	0	= 0	0
A - $b^{\underline{2}}(1)$	1	1	1	1	1	1	1	1	= 255	-1
A - $b^{\underline{2}}(64)-1$	1	0	1	1	1	1	1	0	= 190	-66

Table A-24. Add-Immediate and Subtract-Immediate

The following conditions were tested with carry-in to each respective adder = "zero".

Adder Input Pair	Input Conditions			
	0,0	0,1	1,0	1,1
0		X		X
1		X		
2		X		
3		X		
4		X		
5		X		
6		X		
7		X		

The following conditions were tested with carry-in to each respective adder = "one".

Adder Input Pair	Input Conditions			
	0,0	0,1	1,0	1,1
0		X		
1		X	X	X
2		X	X	X
3		X	X	X
4		X	X	X
5		X	X	X
6		X	X	[illegible]
7		X	X	X

The summary of the test conditions for each adder stage is as follows:

The following conditions were tested with carry-in to each respective adder = "zero".

Adder Input	Input Conditions			
Pair	0,0	0,1	1,0	1,1
0	X	X	X	X
1	X	X	X	X
2	X	X	X	X
3	X	X	X	X
4	X	X	X	X
5	X	X	X	X
6	X	X	X	X
7	X	X	X	X

The following conditions were tested with carry-in to each respective adder = "one".

Adder Input	Input Conditions			
Pair	0,0	0,1	1,0	1,1
0	X	X	X	X
1	X	X	X	X
2	X	X	X	X
3		X	X	X
4		X	X	X
5		X	X	X
6		X	X	X
7	X	X	X	X

The description of the Decimal Adjust Accumulator (DAA) follows:

The eight-bit hexadecimal number in the accumulator is adjusted to form two four-bit binary-coded-decimal digits by the following two-step process:

1) If the least significant four bits of the accumulator represent a number greater than 9, or if the auxiliary carry bit is equal to one, the accumulator is incremented by six. Otherwise, no incrementing occurs.
2) If the most significant four bits of the accumulator now represent a number greater than 9, or if the normal carry bit is equal to one, the most significant four bits of the accumulator are incremented by six. Otherwise, no incrementing occurs.

If a carry out of the least significant four bits occurs during Step (1), the auxiliary carry bit is set; otherwise it is unaffected. Likewise, if a carry out of the most significant four bits occurs during Step (2), the normal carry bit is set; otherwise, it is unaffected.

NOTE: This instruction is used when adding decimal numbers. It is the only instruction whose operation is affected by the auxiliary carry bit.

The Logic Configuration consists of a hardwired add-six in series with each four-bit adder. The input to the enable of the adder is an "or" gate with three required tests "1 & 0" and "0 & 0".

A description of the performed test is as follows:

Vector 197

	Carry	B_7 B_6 B_5 B_4 Hexadec.	Aux Carry	B_3 B_2 B_1 B_0 Hexadec.
Load-157$_{10}$	0	1 0 0 1 = 9	0	1 1 0 1 = 13
1. Set Aux. Carry	0		1	
2. Add 6	0		1	0 1 1 0 + 6
	0	1 0 1 0 = 10	1	0 0 1 1 = 3
3. Set Carry 4. Add 6	1	0 1 1 0 + 6	1	
5. Accumulator = 3$_{10}$	1	0 0 0 0 = 0	1	0 0 1 1 = 3
Vector 201				
Accumulator = 0 3$_{10}$	1	0 0 0 0 = 0	1	0 0 1 1 = 3
5. Add 6			1	0 1 1 0 + 6
6. Clear Aux. Carry			0	1 0 0 1
7. Add 6	1	0 1 1 0 + 6		
	1	0 1 1 0	0	
Accumulator = 105$_{10}$	1	0 1 1 0 = 6	0	1 0 0 1 = 9

In the decimal adjust mode, the input OR gate that enables the decimal adjust accumulator adder was not checked for "0 & 0". That is, the test of "not carry" and "not greater than nine" was not applied to either adder. This test is necessary to ensure correct operation.

In summary during the Arithmetic portion of the test:

1) All op codes assooiated with Arithmetic operations were used.
2) Except for the 0, 0 condition with ripple carry-in = "one" for adder stages 3, 4, 5 and 6, all possible conditions were applied to each adder stage.
3) In the decimal adjust mode, the input "OR" gate that enables the DAA adder was not checked for 0, 0.

It is recommended that the conditions not checked in 2) and 3) above be added.

Logic Functions

The logic function OR and Exclusive OR (EXOR) are checked by changing one bit at a time in the data field. Each register has a single "one" bit in a field of "zero's" as indicated in Table A-25. The accumulator and a register are OR'ed, whereupon the accumulator output is checked for the correct result.

In the next operation, the accumulator is EXOR'ed with the same register. For this operation the accumulator still retains the result of the previous OR'ed operation. Refer to Table A-26. During these operations, the other bits (which are EXOR'd with a zero), are unchanged. B_0 is never checked using a "1 & 0" pattern with the OR or EXOR function. This input condition is required to verify the function.

	Register Bits							
Accumulator and Registers	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
A	-	-	-	-	-	-	-	-
B						1		
C			1					
D				1				
E					1			
H							1	
L		1						
A	1							
(120)			1	1	1			

Table A-25. Accumulator and Register Content for OR, EXOR

OPERATIONS	OR FUNCTIONS	EXOR FUNCTIONS
1	$A + B = A_1$	
2		$A_1 \oplus B = A_2$
3	$A_2 + C = A_3$	
4		$A_3 \oplus C = A_4$
5	$A_4 + D = A_5$	
6		$A_5 \oplus D = A_6$
7	$A_6 + E = A_7$	
8		$A_7 \oplus E = A_8$
9	$A_8 + H = A_9$	
10		$A_9 \oplus H = A_{10}$
11	$A_{10} + L = A_{11}$	
12		$A_{11} \oplus L = A_{12}$
13	$A_{12} + A_{12} = A_{13}$	
14		$A_{13} \oplus A_{13} = A_{14}$
15	$A_{14} + 120 = A_{15}$	
16		$A_{15} \oplus 120 = A_{16}$

Table A-26. OR/EXOR Function Test Sequence

Input conditions that should be applied to check the OR/EXOR functions are outlined in Table A-27. Those conditions that were actually applied are shown in Table A-28.

ALU FUNCTIONS	Input Pair Conditions			
	0&0	1&0	0&1	1&1
EXOR	X	X	X	X
OR	X	X	X	-

Table A-27. Input Conditions Required to Verify OR/EXOR

ALU FUNCTIONS	Input Pair Conditions			
	0&0	1&0	0&1	1&1
EXOR	Bit 6 Not Done	X	X	Bit 0 Not Done
OR	Bit 6 Not Done	X	X	X

Table A-28. Input Conditions Applied to Verify OR/EXOR

The test sequence for the logic functions, "and/compare" are shown in Table A-29. The "and/compare" functions differ only in that the compare function does not store a result. Rotate, OR immediate and EXOR immediate are op codes that are used in the function tests for the first time to condition the data in the accumulator. Coincidentally they are also verified.

OPERATION	Register Bits								REMARKS
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	
A = 198	1	1	0	0	0	1	1	0	
A. (252)	1	1	0	0	0	1	0	0	
A.A	1	1	0	0	0	1	0	0	
A.B	0	0	0	0	0	1	0	0	
A + (124)	0	1	1	1	1	1	0	0	OR Immediate
A.C	0	0	1	0	0	0	0	0	
A. (101)	0	0	1	0	0	0	0	0	
A ⊕ (92)	0	1	1	1	1	1	0	0	EXOR Immediate
A.D	0	0	0	1	0	0	0	0	
Compare A	0	0	0	1	0	0	0	0	Carry F/F Set
Rotate	1	0	0	0	1	0	0	0	Rotate right thru carry F/F
A.E	0	0	0	0	1	0	0	0	
Rotate	0	0	0	0	0	1	0	0	Rotate right thru carry F/F
Rotate	0	0	0	0	0	0	1	0	Rotate right thru carry F/F
A.H	0	0	0	0	0	0	1	0	
Rotate	0	0	0	0	0	1	0	0	Rotate left thru carry F/F
Compare A.A	0	0	0	0	0	1	0	0	Equal, clear carry F/F
Compare A.B	0	0	0	0	0	1	0	0	Equal, clear carry F/F
Compare A.D	0	0	0	0	0	1	0	0	Not equal, set carry F/F
Compare A.E	0	0	0	0	0	1	0	0	Not equal, set carry F/F

Table A-29. "AND/COMPARE Functional Test Sequence

(Continued on the next page)

OPERATION	Register Bits								REMARKS
	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	
Compare A.H	0	0	0	0	0	1	0	0	Not equal, set carry F/F
Compare A. (0)	0	0	0	0	0	1	0	0	Not equal, set carry F/F
Compare A.L	0	0	0	0	0	1	0	0	Not equal, set carry F/F
Rotate	0	0	0	0	1	0	0	1	Rotate left thru carry F/F
A + L	0	1	0	0	1	0	0	1	
A.L	0	1	0	0	0	0	0	0	
Compare A. (255)	0	1	0	0	0	0	0	0	Not equal, clear carry F/F
Rotate	1	0	0	0	0	0	0	0	Rotate left thru carry F/F
Rotate	0	0	0	0	0	0	0	1	Rotate left thru carry F/F
Rotate	1	0	0	0	0	0	0	0	Rotate right thru carry F/F.
Rotate	0	0	0	0	0	0	0	1	Rotate left thru carry F/F

Table A-29. "AND/COMPARE" Functional Test Sequence

The inputs that should be applied to verify the "and" function versus those that were actually used are shown in Table A-30.

	Input Pair Conditions			
	0&0	1&0	0&1	1&1
Inputs Required	-	X	X	X
Inputs Applied	-	X	B_0-B_7 Not Done	Bit 0 Not Done

Table A-30. "AND/COMPARE" Input Conditioning Required/Completed

Branching

The branch conditions were checked twice, once for each logic condition. The flag bits were preloaded prior to each test sequence. Operation of the flag bits were previously checked during ALU operations. Table A-31 shows the branching functional test sequence. All necessary branch conditions were applied, and no additional tests are required.

	PRELOADED FLAG CONDITIONS			

	Sign = 1 Zero = 0
	AUX Carry = 0 Parity = 1
	Carry = 1
			Sign = 0 Zero = 1
			AUX Carry = 0 Parity = 0
			Carry = 0	

BRANCH INSTRUCTION	BRANCH yes	no	BRANCH yes	no
Unconditional	X	not possible	X	not possible
Carry = 1	X			X
Carry = 0		X	X	
Zero = 0	X			X
Zero = 1		X	X	
Sign = 1	X			X
Sign = 0		X	X	
Parity = 1	X			X
Parity = 0		X	X	

(Continued on the next page)

	PRELOADED FLAG CONDITIONS			

	Sign = 1 Zero = 0
	AUX Carry = 0 Parity = 1
	Carry = 1
			Sign = 0 Zero = 1
			AUX Carry = 0 Parity = 0
			Carry = 0	

BRANCH INSTRUCTION	BRANCH		BRANCH	
	yes	no	yes	no
Call & Ret ; unconditional	X	not possible	X	not possible
Call & Ret; Carry = 1	X			X
Call & Ret; Carry = 0		X	X	
Call & Ret; Zero = 1		X	X	
Call & Ret; Zero = 0	X			X
Call & Ret; Sign = 1		X	X	
Call & Ret; Sign = 0	X			X
Call & Ret; Parity = 1	X			X
Call & Ret; Parity = 0		X	X	

Table A-31. Branching Functional Test Sequence

Restart, Interrupt and Halt

Table A-32 describes the last section of the functional test which checks the Restart, Interrupts and hold routines accessible through the uP instructions.

All eight restart locations and the halt and interrupts were verified.

Basically, the restart is applied once at the beginning of the uP function test sequence and once at the end of test. The hold function is applied throughout the entire functional test sequence.

Summary

This test equipment manufacturer's approach to testing is to section the uP into functional areas and test each section independently. This procedure involves grouping the instructions according to the section acted upon, and supplying appropriate operands to insure that the specified instruction is executed correctly.

In general his test set provides a good test for the 8080A. It is recommended that a few additional tests be added as explained in the following text.

INSTRUCTION	OP CODE	PRESENT ADDRESS	NEW ADDRESS
Restart	C7	00A0	0000
Return	C1	0000	00A0
Restart	CF	00A1	0008
Return	C1	0008	00A1
Restart	D7	00A2	0010
Return	C1	0010	00A2
Restart	DF	00A3	0018
Return	C1	0018	00A3
Restart	E7	00A4	0020
Return	C1	0020	00A4
Restart	EF	00A5	0028
Return	C1	0028	00A5
Restart	F7	00A6	0030
Return	C1	0030	00A6
Restart	FF	00A7	0038
Return	C1	0038	00A7
Enable Interrupt (EI)	FB	00A7	----
Disable Interrupt (DI)	F3	00A8	----
EI	FB	00A9	----
HALT	76	00AA HI-2 State	----

Table A-32. Functional Test for Instructions

The registers were treated as random addressable memory bits. The pseudo-walking/diagonal pattern was not carried to completion. It is recommended that additional vectors be added to complete verification of bit independence.

During the Arithmetic Operations all op codes were verified and all overflow conditions tested. It is recommended that test vectors be added to check the 0, 0 condition with ripple carry-in = "one" for adder stages 3, 4, 5, 6. All other possible conditions were applied to each adder stage. In addition, a test should be added for a 0, 0 input condition on the "OR" gate that enables the Decimal Arithmetic Adjust adder.

The logical AND, Compare, OR and Exclusive OR function tests were not totally verified. It is recommended that a 0, 0 input condition be added for Bit 6 for the OR and EXCLUSIVE OR functions, a 1, 1 input for Bit 0 for the exclusive OR and AND functions and the 0, 1 input for Bits B_0-B_7 for the AND function.

All op codes were exercised for the branch, restart, interrupt and hold instructions. Also flag bits were verified during the arithmetic and logic operations.

Other observations made during the evaluation of these vectors are:

1) They do not check any intermediate CPU steps that appear on the output status, data and address lines.
2) Each instruction is treated as a discrete indentity and not as a sequence of instructions.
3) The test vector set utilizes very nearly all of the data buffer memory of the automatic tester manufactured by this automatic tester manufacturer. This could explain why certain tests were not completed or why they chose not to check microprocessor intermediate states.

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	μ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* To be avoided where possible.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL
1776-1976